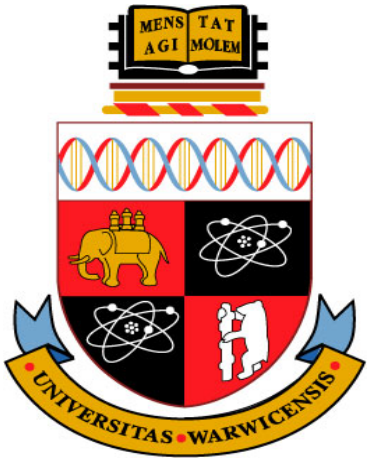


Self Organising Map Machine Learning Approach to Pattern Recognition for Protein Secondary Structures and Robotic Limb Control

Vincent Austin Hall

Submitted to

University of Warwick

for partial fulfilment of the degree of

Doctor of Philosophy

MOAC Doctoral Training Centre

May 2014

Contents

1 Acknowledgments

I would like to thank: my supervisor Prof. Alison Rodger of the Department of Chemistry; Anthony Nash, also from MOAC and Chemistry, who gave lots of advice on the coding side; my collaborator Max J. Ortiz-Catalan of Chalmers University of Technology; my colleagues and friends in MOAC, Chemistry, and Systems Biology. All of these people were great help to me. Thanks to Bonnie Wallace and Lee Whitmore *et al.* from Birkbeck College, University of London; Robert Janes *et al.* from Queen Mary, University of London as I made use of excellent tools they developed, especially Dichroweb; and Narasimha Sreerama and Robert Woody for their CDPro database and publications in the circular dichroism software field. Thanks to my fiancée Anna Cunningham, who was always present to provide lots of support and advice. Thanks to my parents Austin and Heather Hall for making and raising me, and giving me my determination to succeed. Thanks to my son Peter Hall for teaching me to make the most of time available, sleep less and work harder.

2 Declarations

The work contained in this thesis is entirely my own, except where acknowledged in the text, for example the papers that form the results chapters were co-authored with supervisors, colleagues and collaborators. I confirm that this thesis has not been submitted for a degree at another university.

3 Abstract

With every corner of science, engineering and business generating vast amounts of data, it is becoming increasingly important to be able to understand what these data mean, and make sensible decisions based on the findings.

One tool that can assist with this aim is the type of program called a self-organising map (SOM). SOMs are unsupervised Artificial Neural Networks (ANNs) that are used for pattern recognition, dimensionality-reduction of datasets, and can give a visual representation of the data using topology. For this project, SOMs were used to do pattern recognition on circular dichroism (CD) and myo-electric signal (MES) data, among other applications.
To the first of these SOMs, we gave the name SSNN for Secondary Structure Neural Network, as it analyses CD spectra to find structures of proteins. CD is a polarised UV light spectroscopy, it is a useful for estimating structures (conformations) of chiral molecules in solution. In this work we report on its use with proteins and lipoproteins. The problem with using CD spectra is that they can be difficult to interpret, especially if quantitative results are required. We have improved the structure estimations compared with similar methodologies. The overall error across all structures for SELCON3 was 0.2, for CDSSTR: 0.3, for K2d: 0.2, but for our methodology, SSNN, it was 0.1.

Another difficult problem the world faces is that thousands more people every year have limb amputations or are born with non-fully-functioning limbs. Robotic limbs can help people with these afflictions, and while many are available, none give much dexterity or natural movements, or are easy to use. To help rectify the situation we adapted the SOM tool we developed, SSNN, to work as part of a software platform that is used to control robotic prostheses, calling it HASSANN,

Hand Activation Signals, SOM Artificial Neural Network. The system works by performing pattern recognition on myoelectric signals, which are electrical signals from muscles. The software platform is called BioPatRec, and was developed by Max Ortiz-Catalan and his other collaborators. The SOM HASSANN was written by the author, who also tested how well the software works at predicting which robotic limb movements are needed.

4 Abbreviations

ANN – Artificial Neural Network

BMU – Best Matching Unit

CD – Circular Dichroism

CCA – Convex Constraint Analysis

CDNN – CD Neural Network, by Böhm *et al.*

CDSSTR – Variable selection method, an SVD algorithm by Compton and Johnson

CONTIN(LL) – Ridge regression algorithm of Provencher and Glockner,

DSSP – Define Secondary Structure of Proteins algorithm by Kabsch and Sander[1]

EM – Electromagnetic

GA – Genetic Algorithm

GP – Genetic Programming

GUI – Graphical User Interface

HASSANN – Hand Activation Signals SOM Artificial Neural Network, Hall and Ortiz-Catalan

HMM – Hidden Markov Model(s)

HT – High Tension

ICA – Independent Component Analysis, similar to PCA

IT – Information Technology

K2d – or "K2D", a SOM for CD structure determination by Andrade *et al.*

K2D2 – SOM by Andrade and Perez-Iratxeta, inspired by K2d

K2D3 – Later version of K2D2 by same team plus Louis-Jeune

Kohonen Map – Self-Organising Map unsupervised ML algorithm architecture

LCP – Left Circularly Polarised as in LCP light

LINCOMB – Linear Combination least squares method

LOOCV – Leave-One-Out Cross-Validation

MATLAB – Software language and platform developed by Mathworks

MES – MyoElectric Signal

ML – Machine Learning (software that learns from data)

MLP – MultiLayer Perceptron

MLR – Multiple Linear Regression

NN – or ANN for (Artificial) Neural Network NRMSD – Normalised Root Mean Squared Deviation

NRMSE – Normalised Root Mean Squared Error

OS – Operating System

P – Pearson correlation coefficient

PDB – Protein Data Bank

PEM – PhotoElastic Modulator

PMT – PhotoMultiplier Tube

PVA – Population Vector Algorithm

RCP – Right Circularly Polarised as in RCP light

RMSD – Root Mean Squared Deviation

RMSE – Root Mean Squared Error

SELCON(3) – Self-Consistent method (3) by Sreerama and Woody

SOFM – Self-Organising Feature Map, or SOM by Kohonen

SOM – Self-Organising Map by Kohonen

SOMCD – SOM inspired by K2d, by Unneberg *et al.*(similar team to K2d)

SRCD – Synchrotron Radiation Circular Dichroism

SSNN – Secondary Structure Neural Network by Hall *et al.*

SVD – Singular Value Decomposition

TM – TransMembrane

VARSLC – VARiable SELection method by Manavalan and Johnson

VQ – Vector Quantisation

List of Figures

5 List of equations

6 Introduction

6.1 Self-organising map machine learning technique

As the rate of technological development increases more fields of research, business, education, politics etc. have started and continued to produce prodigious volumes of data. IBM estimated in 2012 that there were 2.5 exabytes (one exabyte is 10^{18} bytes) of data created every day[10], and this will only keep accelerating. Actually research shows that technology and scientific knowledge have been growing exponentially for some time, and it is increasing this acceleration.[11] This leads to the conclusion that what is needed is constant development of better tools to manage the ever-growing data.

To that end we need to be sure that the software and hardware we use can constantly grow its abilities for producing meaningful results, and that we have enough data scientists, data analysts, software engineers and I.T. people applied to the task. IBM estimates the world will need 4.4 million data scientists by 2015, and this will only be $^1/_3$ filled.[10]

One of the fields that could greatly benefit from more automation of the understanding of the information coming in from machines is the Biophysical Chemistry work of circular dichroism (CD) spectroscopy. CD uses polarised UV light to determine the conformation of certain molecules (proteins, DNA etc.) before, during and after chemical reactions, or other changes in the environment of these molecules. This work aids in understanding the function of these molecules, which helps with understanding biology, designing new medicines, and making sure that the chemical reactants used are good quality.

The research done in this project had the initial aim of developing something that would help Chemistry researchers understand proteins, and make their work easier, faster, and more accurate by becoming a sort of circular dichroism expert software agent. This software, called SSNN for Secondary Structure Neural Network, was designed to do pattern recognition on the CD spectra. The desired outcome was for SSNN to learn what spectral features in the CD spectra lead to which amounts of which secondary structures in the molecules. Though this is not shown to users, it is just understood by the software.

The initial aim was to help CD practitioners by developing a software application to learn about CD, however, the software developed with this aim has grown far beyond that to include: finding the healthiest, comfortable insoles for diabetes sufferers; controlling robotic prosthetic arms and hands, and even matching Chemistry academics for research projects. CD, myoelectric signals for robotic limb-control and diabetes was done by the author, the matching of Chemistry academic primary investogators was done by Alison Rodger. This software is now packaged in a graphical user interface (GUI), along with a guide, so that anyone may download it from the Rodger Group website and use it on their dataset.

As a machine learning method (see section 6.1.8), SSNN, the key software package developed in this work, can be applied to any data set. It is the type of software that will work on unlabelled data, where very little or no prior knowledge is present (data mining). SSNN is a self-organising map artificial neural network, or simply SOM, the SOM technique was invented by Teuvo Kohonen circa 1982.[12] It is also referred to as a self-organising feature map (SOFM), or Ko-

honen Map. The acronym SOM is used to refer to an algorithm that constructs an organised map, and the map itself. Hence the term self-organising map; it is auto-organised. The self-organising map algorithm is a data-analysis technique that works automatically, in an unsupervised manner.

This approach was employed to estimate structures (or conformations) of proteins using circular dichroism (CD) spectroscopy, and later to control robotic prosthetic limbs by performing the pattern recognition stage of interpreting myoelectric signals (MES) from patients.

This work focuses on unsupervised learning, and the SOM/SOFM or Kohonen Map. The SOM places features of the dataset on its map in a way that enables the user to gain knowledge about the features present, and their relationships with other features, from the location on the map.[13] The way the SOM is organised to cluster data and display it in a 2 dimensional form helps one to easily get an impression of what the topographical relationships between elements of the data are, especially high-dimensional data that is very hard to picture or make clear, quick conclusions about.

In summary, a SOM takes input vectors and generates a clustered map representation of them, with interpolations. The interpolations are calculated as a weighted sum of the few nearest nodes that best represent the experimental vectors; these nearest nodes are called the best matching units, or BMUs. These BMUs are closest in Euclidean distance. The Euclidean distance depends on differences between two vectors at the same element, see equation 1.

$$d = \sqrt{\sum_{i=1}^{n}(x_i - x_i')^2} \tag{1}$$

where x is the observed datum in the input vector at position i, and x' is the calculated or theoretical value of it from the model generated by the SOM. Lower case n is the number of elements in the input vector.

For the complete training of a SOM, the following is repeated for thousands of iterations:

1. A CD spectrum (another input) is selected at random from the database or reference set, and compared with the map to find a BMU (Figure 1).
2. The BMU on the map is made more similar to the input vector.
3. The neighbours of the BMU are updated too (Figure 2).

Through this process the SOM makes an organised map, or a clustered map of input vectors. Similar vectors should be close to each other, and dissimilar ones further apart. See Figure 1 for a view of how inputs correspond to the nodes of a SOM.

Figure 2 shows the neighbourhood of a BMU is selected for learning as well, while the rest of the map remains static for this iteration. This is how the clustering works, if it does not make the BMU neighbourhood similar to the BMU, then there will never be regions where certain classes of vectors are clustered, just a random collection of un-clustered vectors.

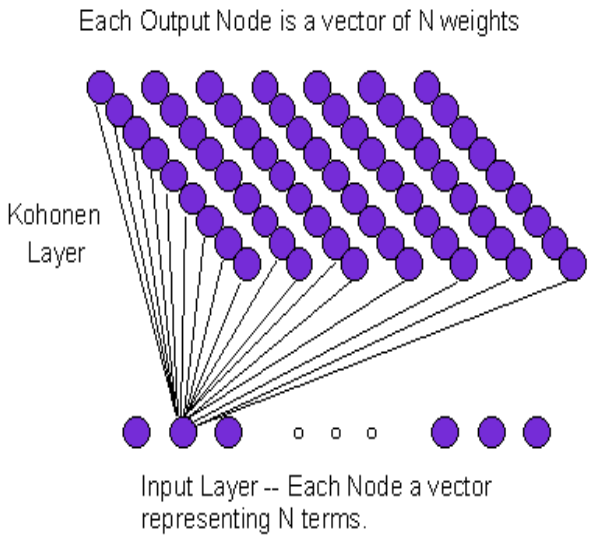


Figure 1: Schematic illustration of a SOM. Each input vector is presented to the map, and each node comes to represent an input vector, or the interpolation between inputs. This assignment is completed by the time training has finished.[2]

See Figure 3, for an example of what one level of a SOM looks like once the data it holds have been clustered. There is one level for each dimension of the data space. The red, higher areas represent the regions of the data space that have more positive values at this, the n^{th} element of each input vector. Note how each value has been located amongst its equals or near equals. However, there are still two areas with larger negative values: the low region in the front–left, and the very low region in the front right. This shows that they belong to input vectors that are of two different types. The other elements of the input vectors (those not in this layer) place them in different classes, even if the elements on this layer are similar. This is because there are far more relationships between data points in the vectors present than just this layer, so they have a much stronger influence on the organisation of the SOM than the elements of vectors that are shown here (this layer).

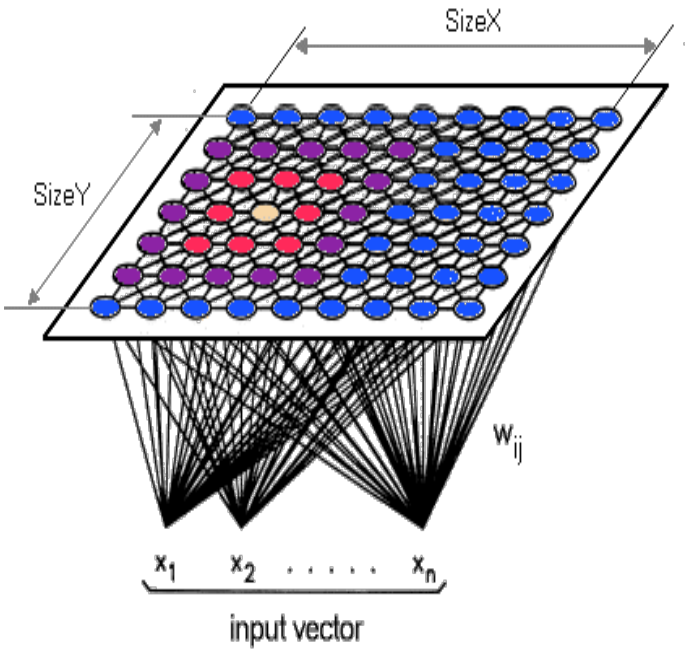


Figure 2: Schematic of the SOM training process. The BMU (cream circle), and its neighbourhood are selected for training to become more like the input data, with those closest to the BMU learning more in the current iteration than those further from it. The rest (blue circles) are not updated this iteration.[3]

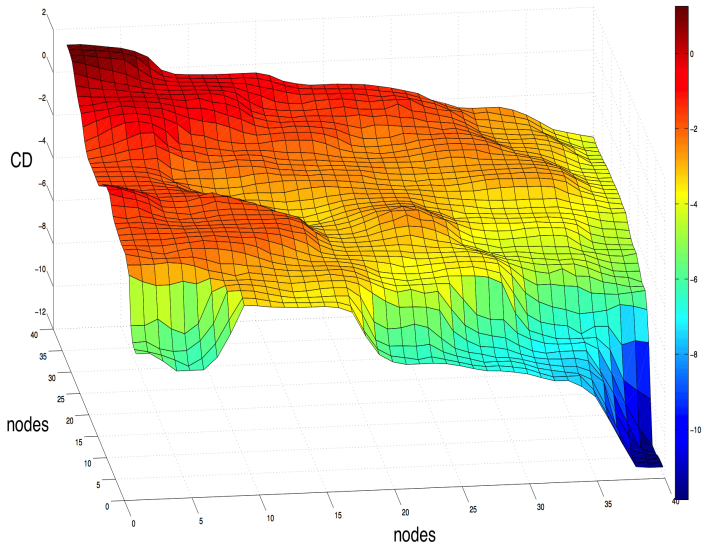


Figure 3: The trained SOM. The example here is a view of one level of a SOM that has clustered CD spectra; this represents the light intensities at one wavelength. Note how the terrain undulates smoothly, forming clusters: hills and valleys.

6.1.1 Origins of SOM

SOM is similar to the simpler, older methodology called VQ, vector quantisation. The operation VQ performs is to compress and map continuous or discrete data for transmission or storage in a digital channel. Vector-valued input data space is segmented into a number of adjacent regions. A single model represents each region optimally (a single point on the map). This is called the codebook vector. In a SOM the equivalents of codebook vectors become globally ordered in space as well.[14–16]

6.1.2 Training a SOM

There are two ways to train a SOM 1) stochastically selecting individual examples from the dataset, or 2) batch training, by considering all reference set examples in one batch, and all models are updated simultaneously in a single step. Stochastic training needs at least a few thousand training iterations to make sure all examples from the reference set have had a good chance to have their impact on the map. Batch training only requires a few dozen iterations before training is complete, this is because a batch will typically contain many individual vectors, so one iteration represents many from a stochastic training point of view. Batch training is usually quicker and more likely to converge. Batch training does not require any learning-rate parameter either (see sub subsection 6.1.6). The stochastic training requires two learning-rate parameters, L_0, the initial learning rate, and k_1, the rate at which learning decelerates. L_0 should be a small number, in this work 0.06 or occasionally 0.1 were used. This ensures the algorithm learns a bit each iteration, if the L_0 were 1.0, then a node would become exactly the training set spectrum in one iteration. This is not idea for clustering, it doesn't come to as good a convergence as slower learning. The k_1 parameter should be very small, as it multiplies the exponential in the learning rule equation, and learning is needed throughout training.

6.1.3 Validating a SOM

One reasonably reliable way to validate the training parameters is to use leave-one-out cross-validation (LOOCV). Here the SOM is trained with all but one of the training set vectors, and tested on the vector that was left out of training. This process is repeated until each of the training vectors has acted as the test vector. Training parameters of the SOM were validated using LOOCV; the size

of the map, the learning rate, the number of BMUs that go into making the model, the learning equation parameters t_1 and k_1, and others. This is detailed in "Paper 1".[17]

6.1.4 Applications

Applications of SOM for clustering data are wide and include exploratory data analysis, control systems, telecommunications, finance, natural science, statistical analysis, biomedical analyses, profiling of criminal behaviour, characterisation of galaxies, categorisation of real estate, and linguistics/organisation of texts.[18–23] Some of the largest applications are bioinformatics, and huge textual databases. In 2012, Kohonen *et al.* found over 10,000 applications of SOM[14]. Besides SOM, there are other artificial neural networks, and there are many other architectures, including evolutionary programs, artificial immune systems, random forests, Bayesian networks, here the subject will be SOMs.

6.1.5 Good practice with SOMs

The distance metric to measure the distances between data vectors on the map needs to be decided upon. This could be Euclidean or more generally Minkowski, Euclidean distance being Minkowski in 2 dimensions.[14]

The SOM is usually a square or hexagonal array of nodes containing the data vectors. The square array is easier to build, and the hexagonal is better for visualisation purposes. It has been suggested that one should use a cyclic array, such as a toroidal or spherical shape so that the edges of the map loop around and become contiguous with each other. This is because there can be irregular spacing between adjacent models at the extreme edges of the map. This format is only suitable if the data itself is cyclic in some way.[14–16,24]

The size of a SOM should be large enough to include models for all spectra or data vectors in the reference set, and to extract fine features of the data. So if the data are expected to contain many fine features, a large map should be used, otherwise the size should not be very large to optimise computational time. The best way to establish the optimum size is trial-and-error. Typical sizes are in the order of a few dozen to a few hundred nodes, and the dimensions of an array should approximate the two principal components of the input data. Some suggest using 4 to 10 nodes per expected class in the dataset.[25] However, this should be validated by training a map of a particular size, and comparing its results with results of maps of different sizes.

To initialise a large map, it may be advisable to train a small map, initialised with the principal components, then once trained to a partially complete state, to add nodes in-between those existing, which take values interpolated nonlinearly

from the existing nodes. After that the larger map is trained until it reaches acceptable clustering.[14]

When searching for the BMU for any reference or test vector in general, one may save a great deal of computational time by taking note of the locations of the previous BMUs, and the corresponding training data. In subsequent iterations the search for the BMUs can be restricted to the vicinity of the previous winners. This may also be done by pre-training a smaller map first: the approximate regions of likely BMUs can be given to the large map. So if a cluster of one class of data (Class 1) is found in the top, right corner, and another cluster of different class of data (Class 2) is found in the opposite corner, then it can be assumed that these two clusters will be in opposite corners of any map with the same data. Training a larger map with these data should then proceed by only looking for BMUs in opposite corners for these inputs when they are selected. One can search only in the top, right corner for BMUs for class 1 input, then that corner will be trained on that class 1. Later, when Class 2 data requires a BMU corner, the bottom, left corner could be assigned to that. This is manually biasing the larger map, so injecting knowledge, but that knowledge comes from the earlier clustering with the smaller map.

Another method to speed up computation, and reduce memory requirements is to truncate the data vectors:[14] for example, selecting every fifth element in the vector can reduce computation by a factor of 5, without losing many of the features.

On may also select features one thinks are important in each input vector, and then make a new dataset, where the first column is Feature 1, column 2 is for Feature 2, etc. This can greatly reduce the data, and therefore training time,

while hopefully keeping most of the important information. Features used should be theoretically sound, and should be tested to determine if they lead to good predictions, if there is a desired outcome.

6.1.6 SOM applied to circular dichroism spectra

In our methodology, SSNN, the spectra are clustered in an unsupervised manner, but the protein secondary structures corresponding to these spectra are known, so the algorithm can be improved until a certain level of accuracy is attained. This is done by running the software many times (validation, usually LOOCV) while varying the iteration count. An error metric that is useful in this effort is the normalised root mean squared deviation, or NRMSD. The formula for it is the same as the root mean squared deviation divided by the range of the observed data:

$$RMSD = \sqrt{\frac{\sum (x_i - x_i')^2}{n}} \tag{2}$$

and

$$NRMSD = \frac{RMSD}{x_{max} - x_{min}} \tag{3}$$

where for the CD data, x_i is an element of the experimental circular dichroism, CD, spectrum; x_i' is the estimation of x_i from the predicted spectrum; n is the number of elements in the spectrum (for example, for us this was 51 due to the range of CD data wavelengths: 240 nm-190 nm); x_{max} and x_{min} are the maximum and minimum of the observed values. The NRMSD is used in Dichroweb to compare the CD analysis methodologies or algorithms available for use on that website. On Dichroweb, each methodology used produces an NRMSD for the spectrum of each test protein given to it. This is intended to be a guide as to the accuracy of the structure predictions made by that methodology, for this specific

test CD spectrum.

When testing many different SOMs with various numbers of iterations, using a LOOCV test, the SOM arrangement with the smallest NRMSD for the test spectrum in question is considered to have the best number of iterations.

A summary of how a SOM works is that one starts with a map, a collection of random vectors arranged in an NxN square or hexagonal lattice. This map comes to represent the data. This will be expanded with much greater detail later.

In our case we made a 40x40 square map of CD spectra. So the map in 3 dimensions is 40x40x51. Various sizes of map were tested, this was the best, as is detailed in Paper 1.[17] The value of 40x40 nodes in the SOM trained with our then 48 CD spectra was arrived at by running LOOCV for various map sizes. It was found that a larger map size generally produced smaller error, but larger than 40x40 nodes did not improve the accuracy much, and took a lot more computational and real time to train. We found that this very large map, with 5 or 6 BMUs to make the model spectra from produced models with elements from each of the BMUs: different regions of the spectrum were 'inherited' from the BMUs. If this were a simple classification question, then having a small map with very few models might have been successful, but we were looking for 6 numbers adding to 1.00 for the 6 different structure types, and the sets are not boolean, but fuzzy. So for the most part each protein contained non-zero values for all 6 structures in varying amounts. The reference or training dataset also contained these, and the test set did not contain any of the same spectra, so the structure results should not be the exact structures of one model spectrum. Also, when a small map is being used, making a model out of several BMUs tends to result in the model inheriting characteristics from most of the nodes, or potential BMUs

in the map. For example, having a map with 9 nodes, and using 6 of those to make the model of the test spectrum (6 BMUs) would be illogical, because the model would be very average every time, there would not be much room for individuality, and making a different decision (structure vector) each time.

The 48 spectra were later expanded to 53. We added 2 theoretically 100 % α-helix proteins, by taking two helix spectra and stretching the peaks until they were where a protein of 100 % helix would be. We added 3 truly 100 % random coil protein spectra. We present the map with the reference set of 53 CD spectra in the sequential, stochastic manner detailed above. SSNN selects a spectrum, and tries to find a match for it on the map. Each vector on the map is compared with the selected CD spectrum, and the most similar one is said to be the BMU. A learning rule (equation 4) makes the random vector more similar to the CD spectrum. The region around the vector or BMU on the map, the neighbourhood, is made more similar to the CD spectrum by a smaller amount: decreasing toward the edge of the neighbourhood (Figure 2).

The SOM architecture is used by some other methodologies for determining protein secondary structure from circular dichroism, e.g. the K2d family[26–28], and SOMCD[13]. The SOM is a good artificial neural network, ANN, for CD spectra, as it allows one to see the clustering of the spectra without too much difficulty, as each node of a SOM trained with CD spectra contains a spectrum; the SOM is topological. The SOM is not the only ANN that might be used for pattern recognition of CD spectra, but the visualisation capabilities do make it appealing for such work.

After the SOM has clustered the CD spectra so similar spectra are close, it does the same with the secondary structures that correspond to the CD spectra.

These structures come from X-ray crystallography data. SSNN makes a map of structures that correspond to the spectra by finding the CD spectra in the map that are most similar to the reference set spectra. There are also many spectra that are intermediate between the training set BMUs, and these are referred to here as virtual spectra. These virtual spectra number (40x40 - 53 = 1547), more of this is covered in Paper 1.[17]

Briefly, the 53 training or reference set spectra take up 53 nodes on the map, the other 1547 nodes hold spectra intermediate between those. These are virtual spectra, and are hybrids made from the elements from their training set neighbours.

During the training process, learning proceeds so as to cause the SOM to learn fast at the beginning, and exponentially slower towards the end of training. The size of the neighbourhood is also reduced in size with iterations. The equation for the learning is:

$$L(t) = L_0 \cdot exp^{(-k_1 \cdot t)} \tag{4}$$

where L is learning rate, L_0 is the initial learning rate, t is the current iteration, and k_1 is a parameter of size $<< 1$. According to Kohonen,[14] the learning rate of the nodes should decrease monotonically (e.g. hyperbolically, exponentially, or piecewise linearly) with iterations.

The equation for the neighbourhood radius is as follows:

$$radius(t) = \begin{cases} (RADIUS_0 - 1) \cdot (1 - \frac{t}{t_1}) + 1 & \text{if } t \leq t_1 \\ 1 & \text{if } t > t_1 \end{cases} \tag{5}$$

where $RADIUS_0$ is the initial radius of the neighbourhood, t_1 is a parameter about a third the size of number of total iterations.

According to Kohonen, the first 1000 or so steps of training the SOM should be for ordering the map topologically, the rest of the time should be spent carefully ordering the models to reach their best states. This may take 10 times longer than the initial topological ordering stage.[14]

6.1.7 Developments to SOM

Fritzke 1994[29] made a self-organising network inspired by Kohonen's SOM that adds more cells or nodes with iterations, i.e. "Growing Cell Structures". There are a few main differences between this and the SOM approach:

1. The learning or adaptation parameter remains fixed, while in a SOM it reduces monotonically.

2. Only the BMU and its direct topological neighbours are updated to become more similar to the relevant input vector, not a whole region.

3. Cells can also be removed

Removal of a cell can happen when the cell is considered superfluous, i.e. if it has a position in a region of the data space with very low probability density; lower than a threshold. The removal causes there to be fewer connections between cells, and there can even be separate clusters with no links to each other, see Figure 4.

This method works using Voronoi regions, or regions created using Voronoi tessellation.[30,31] Voronoi regions result from clustering data, such that every point in a Voronoi region of a Voronoi diagram is closer to the centre of each region than any other centre. The centre of a region is called the seed, or centroid, and the regions are known as Voronoi cells. This method is named for Gregory Voronoy,

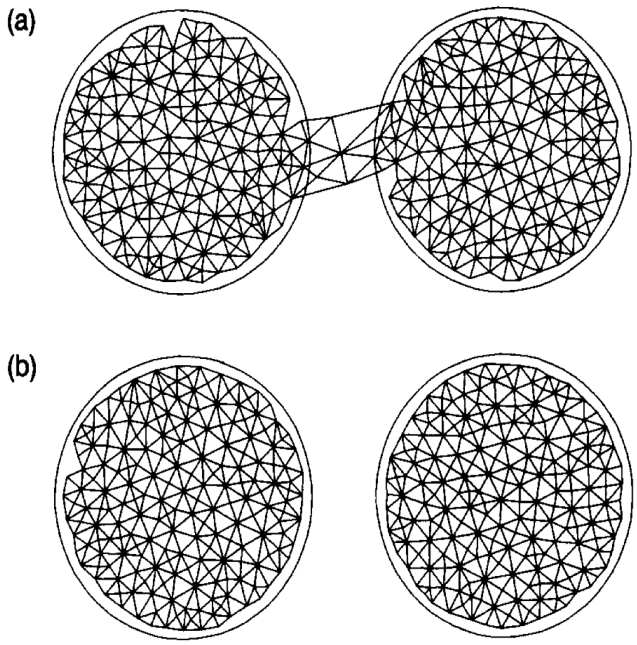


Figure 4: Voronoi regions, a) a well adapted network arrangement with good modelling of a probability distribution the connections are mostly short. This is good, but there are still some extraneous connections. In b) these extraneous connections have been removed, and there are only short connections. The network now models the probability distribution nearly perfectly

and an alternate name is Dirichlet tessellation, after Peter G. L. Dirichlet.[30,31]

The map is made of k-dimensional hypertetrahedrons, which look like triangles when plotted in 2-dimensions. So the deletion of cells or nodes must always leave behind connections forming hypertetrahedrons (in high dimensions, or tetrahedrons in 3-dimensions, or triangles in 2-dimensions), rather than just lines connecting free-floating nodes. The connection lines, or edges represent neighbourhood relationships.[29,30]One advantage of using this 'growing cell structure' methodology is that it makes obvious separation between different classes, as there are no connections between nodes of different classes, due to deletion of cells (nodes).

Fritzke *et al.* gave an example of classifying animals using the frequency cell approach, with data taken from a SOM paper by Ritter and Kohonen.[29,32] The SOM classified the animals correctly, but borders between the different classes (bird, herbivorous quadruped, carnivorous quadruped) had to be drawn in by a human, while the self-organising growing cell structure of Fritze classified correctly, *and* showed the boundaries automatically (projected in 2-dimensions).

In 1997 Kohonen *et al.* developed a SOM algorithm for signal processing called ASSOM, or adaptive subspace SOM.[33] In ASSOM, each node of the map adapts to become an expert for one class of transformations. ASSOM makes statistical representations that refer to various temporal events. As a member of the SOM family of algorithms, it works by competitive learning, where there is only 1 winner. This is in contrast to PCA, for example, which finds average data features that come from global properties that are not temporal.
The aim of the ASSOM algorithm is to solve the long-standing problem of recog-

nising patterns that are simple transformations such as rotation, scaling, and translation. Examples they looked at were speech waveforms, and coloured noise patterns from photographic images. ASSOM was successful in finding relevant filters.[33]

6.1.8 General good practice in machine learning

SOM is a neural network type of machine learning; this section describes what machine learning is, and how best to use it.

Machine learning is the branch of computing that deals with software that can adapt its models to datasets; it learns from the data to produce more accurate predictions. There are two main types of machine learning: supervised (for labelled data) and unsupervised (for unlabelled data). An additional method is reinforcement learning, described just below.[34]

In supervised learning, the algorithm is given the input vectors, and their target vectors, so that, during training, it may perfect its patterns to replicate those targets/predictions. Examples of supervised learning include multilayer perceptrons (MLPs), and genetic algorithms (GAs) with definite goals.[29]

In unsupervised learning, the algorithm will study the data, and find patterns that are previously unknown. This process includes clustering, for example SOM, hidden Markov models (HMM), and GAs with open-ended goals. SOM is an unsupervised, competitive learning method.[14,29,35] The ability to use unlabelled data makes unsupervised learning useful for data mining, where the truth is not known until correlations and patterns are found by the software, usually

because no human can understand the data, as they are far too high dimensional, so there can be no correction of the predictions.

There is also reinforcement learning, which is a method of programming software agents using rewards and punishments, while not requiring an explanation of how to perform a given task. The agent must learn using trial-and-error methods in a dynamic environment. The two main ways to solve reinforcement problems are 1) to search for a behaviour that performs well in the environment, and use it (GAs and GP), and 2) to estimate the usefulness of taking actions in the environment by using statistical techniques and dynamic programming techniques.[36,37]

The basic goal of all machine learning, ML, techniques is to gain the ability to generalise to much more than just the training set. This is because the ML system will never be able to have all information about every possible situation, however, just having vast amounts of data does not suffice to be able to generalise well. Like a brain needs to be able to learn new knowledge, because instinct will never give it the ability to cope with every possible situation in a changing world.

If we give a ML system a Boolean function (two possible correct answers for each variable: true or false) with a moderate 100 dimensions, and one million examples in the sample data, this would mean that there would be 2^{100} - 10^6 examples with unknown classes, i.e. approximately 10^{30} unknown classes. The way to solve this problem, or set of problems, is to include some knowledge or at least assumptions along with the raw data. 100-dimensional data is easily reached, the circular dichroism spectroscopy data we work with is 57 dimensional, and the myoelectric signals we use for robotic limb control are 63 dimensional.[4,9,17,38]

Wolpert gave a name to the problem of having to include knowledge, rather than just data, to make better learners. He called this 'no free lunch' theorem, meaning that, when trying to understand all functions that are needed in the universe, on average no learner may do better than guessing randomly. In this sense there is no difference between algorithms without knowledge.[39–41] No algorithm can predict everything in the universe. It will need some knowledge to be better than another prediction algorithm that has no knowledge, just data. This may lead to the problem of having to supply vast amounts of data to train the algorithm, assuming the functions that describe the patterns in a given data space are uniformly distributed.

Fortunately, the functions that describe the samples are not taken uniformly from all possible mathematical functions. Assuming the following: that similar examples will have similar class membership, that the functions will be smooth, and that complexity is finite, are all in the set of few assumptions that are sufficient to understand rather well.

When learners use induction, a little input knowledge can be transformed into a lot of output knowledge. It is superior to deduction in that it requires a much smaller supply of knowledge to show good results.[41] However, it was always believed that with inductive reasoning the correct decisions and causations cannot be guaranteed, unlike with deductive reasoning, where they can be. More recently this has been found to be untrue; there can be guarantees on results gained from inductive reasoning, especially when accepting probabilistic guarantees.[41]

In any machine learning methodology, before beginning the training, the data need segmenting into training set, validation set, and test set. This is so the

classifier may a) learn the data well, b) have its parameters and architecture validated, and c) be tested on a new dataset. The testing phase is needed so it may show that it can make general conclusions about data, and is not over-trained on specific data. This cannot really be done when the answers are not known, as can be the case with data mining.

Having the test dataset enables one to answer the question of how accurately the classifier needs to predict the target vectors; too little training, and the algorithm does not know the data, and has not recognised any patterns (underfitting), too much training on the specific training set, and the algorithm cannot generalise (overfitting).[41]

Underfitting is where a learning or heuristic algorithm has not been trained enough to recognise patterns in the dataset, and so cannot make good predictions or classifications for this or any dataset. Overfitting is where a heuristic algorithm has trained too much on one particular dataset, and believes that is all there is to know, so it cannot predict or classify any new data well. It keeps assuming the new data is exactly the same as, or very similar to the training data set.[41]

One can understand the problem of overfitting by dividing the error of generalisation into two parts: bias and variance.[42] Bias is learning the wrong thing consistently, and variance is caused by a tendency to learn random things, and not the true information.

There are various ways to work against overfitting. Cross-validation can help, but fitting too many parameters with it can also cause overfitting. Regularization

terms added to the evaluation function are also a good idea, like penalizing the size of classifiers so that they do not get too large and have space to overfit.

The next biggest problem for machine learning is "the curse of dimensionality". As dimensions (or number of features) of the data grow, generalising becomes exponentially more difficult. If the number of features in the database is again just 100 with a Boolean solution space (two options: true or false), and even as many as 10^{12} samples are given to the classifier algorithm, the samples still only cover 10^{-18} of the input space ($2^{100} \approx 10^{30}$).

A more deep-rooted and important issue is that the way ML algorithms classify based on similarity fails in high dimensions. When there are 100 dimensions, and only 2 are being compared, the noise from the other 98 dimensions can act as noise, and can drown-out the signals from the first 2.[14]

So is gathering more features always a good thing? No, it can be that they offer nothing new that is not already known, and their pros can be outweighed by their cons, when they contribute to there being too many dimensions to the features. Assuming the examples are uniformly distributed, then the detriments to having more data become much worse quickly. Actually, the situation is not as bad as it may seem, as examples are *not* usually spread uniformly, but cluster about the lower-dimensions.[14]

In success or failure of ML techniques the most vital element is that the features used are good. When dealing with raw data, learning is not straightforward. However, if one forms good features from the raw data, learning can proceed more easily. This is usually a reason for the greatest effort in ML. If

the ML algorithm takes raw data, it will find pattern recognition far more difficult than if it works with data that has had some important features highlighted.

Machine learning is an iterative process of 1) applying the data or features to the learning algorithm, 2) analysing the results, 3) altering the algorithm and maybe the data. Here learning is usually the easy stage (ML algorithms have general application, so easily learn data), while feature engineering is the most difficult stage, as it is domain-specific. Automating feature engineering to an increasing degree is therefore a very worthy pursuit. This could be done by creating a large collection of features that might be useful and picking those that help best to improve classification information.

Despite all of the above, an algorithm with vast amounts of data still wins over a better classifier with much less, which of course might lead to computation time issues.[41]

There were no computation time issues for the CD project, as the dataset that trained the selected ML technique, the SOM, was 57 by 53 data (points), from 57 dimensions and 53 examples. As mentioned above, the aim was to take the CD spectra and structure knowledge, and do some pattern recognition on them to find spectral features that link the spectra to the secondary structures. The question was: what features lead to what structures of proteins, and in what proportions? Aspects of the history of this field of work, along with some chemistry and physics background knowledge on CD are summarised below.

6.2 Circular Dichroism for protein secondary structure estimation

Circular dichroism, CD, is a spectroscopy that has been used since the 1960s in structural biology to study the structures of peptides, polypeptides and proteins.[43]

CD was discovered in the 19th century by Jean-Baptiste Biot, Augustin Fresnel, and Aimé Cotton.[44] CD is the difference in absorption of left- and right-circularly-polarised light (LCP light minus RCP light), see the equation below[5]

$$\Delta A = A_L - A_R \tag{6}$$

where ΔA is the difference in light absorbed by the chiral molecule, and A_L and A_R are the absorptions for the LCP, and the RCP light respectively.

Absorbance is the logarithm of the incident divided by the transmitted radiation, as in the equation below:

$$A = log_{10}(\frac{I_0}{I}) \tag{7}$$

where A is absorbance, I_0 is the incident radiation, and I is the transmitted radiation.[45–47]

In order for a CD spectrum to produce a useful, non-zero signal, the molecules have to be in a suitable solution (known as a buffer), and the molecule must have some chirality (explained just below). A buffer is a solution that resists changes in acidity, so if a small amount of acid or a base is added to the buffer solution that contains the protein being studied, then the acidity (pH) should remain approximately the same. A good buffer should be transparent in the wavelength of light

that is being used. A chiral molecule cannot be superposed onto its mirror image, the arrangement in space is very similar, but not exactly the same, just in the same way that a left and a right hand are not arranged in exactly the same way, and are not simply reflections of each other. CD is based on the Cotton Effect or Optical Rotatory Dispersion, which is where the biased absorption between LCP and RCP light by stereoisomers of a chiral molecule causes redistribution of electrons in a helical way, for certain transitions, at particular wavelengths. By studying the CD spectrum of a chiral molecule, the handedness of the changes in electron positions can be found. Some aspects of the structure of the molecule in three dimensions can be derived from this.[43,48] Stereoisomers are molecules that have the same atoms and sequence, but different spatial arrangements.[49,50]

The basis of the CD spectrum is electronic transitions, this gives rise to spectral features: peaks and negative bands. In Figure 5, we can see the CD spectral features that arise from the structures present in a chiral molecule. α-helices have a characteristic large peak at about 190 nm, and two negative bands at 208 nm (called the $\pi \rightarrow \pi^*$ transition) and 222 nm (n $\rightarrow \pi^*$ transition). β-sheets are characterised by a peak between 195 nm and 202 nm ($\pi \rightarrow \pi^*$ transition), and a negative band between 215 nm and 220 nm. Turns are negative at 180 nm to 190 nm, and positive at 200 nm to 205 nm. A disordered structure has a negative band at 200 nm. These are for the same set of electronic transitions, they just shift in wavelength a little for different structure types.

Electronic transitions involve electrons jumping energy levels near the ground state. The n $\rightarrow \pi^*$ transition is when the electron in the ground state transfers from the n molecular orbital to the π^* molecular orbital. The $\pi \rightarrow \pi^*$ transition results from a π ground state electron jumping up two levels to the π^* level.[5,17]

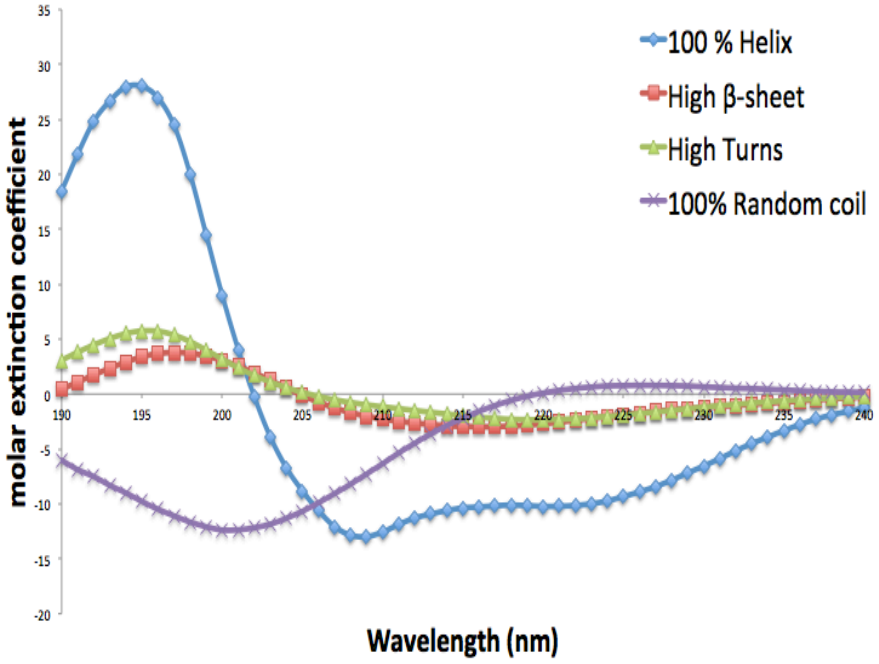


Figure 5: The different CD spectra that are produced by the various structures of chiral molecules.[4]

CD can be measured in terms of molar extinction coefficients, or $\Delta\epsilon$, also called molar absorptivity, which has the units L mol^{-1} cm^{-1}:

$$\Delta\epsilon = \epsilon_L - \epsilon_R \tag{8}$$

and

$$\Delta\epsilon = \frac{\Delta A}{cl} \tag{9}$$

where ϵ_L and ϵ_R are the molar extinction coefficients for the LCP and RCP light, c is the concentration of the chiral molecule in solution measured in mol·l^{-1}, and I is the cell pathlength, measured in centimetres. The pathlength is the width of the (usually quartz or glass) vial that containing the sample, which the light

passes through.

We worked with $\Delta\epsilon$ values, which are per mol, so our data are independent of the number of residues. Equation 9 is derived from Beer Lambert's Law:[5]

$$A = \epsilon cl \tag{10}$$

$\Delta\epsilon$ values can be converted to mean residue ellipticity (MRE or θ) using the simple equation:

$$\theta = \Delta\epsilon * 3298.2 \tag{11}$$

MRE is measured in degrees cm^2 $dmol^{-1}$ $residue^{-1}$. CD spectrophotometers output the CD signal in units called millidegrees. This is because, historically, the change in polarisation of linearly polarised light into elliptically polarised light passing through the sample was measured as the CD signal. Millidegrees can be converted into $\Delta\epsilon$ units using the equation below:

$$\theta/millidegrees = 32,982 \cdot \epsilon cl \tag{12}$$

which is the Beer Lambert Law (equation 10) restated for different units.[5]

LCP light is defined as: when viewed from the source, the electromagnetic field of LCP light rotates in an anti-clockwise direction, and the EM field of a beam of RCP light rotates in a clockwise direction.[51]

6.2.1 Where CD signals come from

Circularly polarised light has a magnetic dipole moment that rotates (a magnetic dipole transition moment, $\mathbf{m}$), and an electric transition dipole moment ($\boldsymbol{\mu}$) that oscillates linearly (back and forth). These two moments combine to form a helical

motion of the electromagnetic field. The helical motion of the EM field causes an electron to move in a helical path, as can be seen in Figure 6.[5,6]

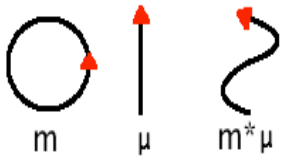


Figure 6: The magnetic dipole transition moment, **m** exhibits circular motion, and the electric dipole transition moment, $\boldsymbol{\mu}$, exhibits linear translation. These combine to make helical motion of the EM field.[5,6]

So the EM field moves in a helical motion, but how do CD practitioners find the signals these helical motions of light produce? The CD machine, which is called a spectrophotometer or spectropolarimeter records the CD spectra. An extremely important part of a circular dichroism spectropolarimeter is the component that makes sure the machine produces exactly equal intensities of both LCP and RCP light. This component is the PEM, or photoelastic modulator. The UV light source is a xenon arc lamp, which also emits visible light. Various mirrors, prisms and slits are needed to collimate the light from the lamp. Subsequently, the light is linearly polarised (as would be useful for linear dichroism), and the PEM circularly polarises the light.

The PEM is made of crystalline quartz stuck to a piece of isotropic quartz (silica). Light travels through the silica section. The light is split into two orthogonal beams, and due to the birefringence in the PEM, the two beams experience different refractive indices. An alternating current, AC, is applied to the crystalline

section of the PEM, this makes it oscillate at 50 kHz. Varying the amplitude of the voltage causes the PEM to select alternately for light polarised in one direction, followed by the orthogonally polarised light.

So the PEM effectively forms a wavelength-dependent quarter-wave plate that is needed to make circularly polarised light. In Figure 7, see a diagram of the optics of a CD spectrophotometer (another name for a CD machine). The PMT (photomultiplier tube) detects the light that was not absorbed by the sample, and converts it to an electrical signal; it vastly multiplies the current received, and is a very sensitive detector.

The CD signal is then determined by the ratio of the AC to DC elements detected by the PMT, see equation 13. The sign of the CD derives from the phase of the AC element using a lock-in amplifier that uses the AC voltage of the PEM as a time reference.[5,36]

$$CD = \frac{< AC >}{DC} \tag{13}$$

Instruments can now hold the DC current constant, with the use of a servo that adjusts the PEM voltage. With a constant DC voltage, the CD is proportional to the <AC> voltage.

6.2.2 Buffers

To perform CD analysis of chiral molecules, an appropriate buffer should be used. A buffer is a solution containing a weak base and its conjugate acid, or a weak acid and its conjugate base. The purpose of a buffer is to resist changes in pH when small amounts of acid or base is put in it[52]. A buffer should be at a concentration in the solution such that it may withstand variations in the pH due to the addition of a highly charged ligand, for example ATP^{4-}. The buffer must be used within approximately 1 pH unit from the appropriate pKa, the acid dissociation

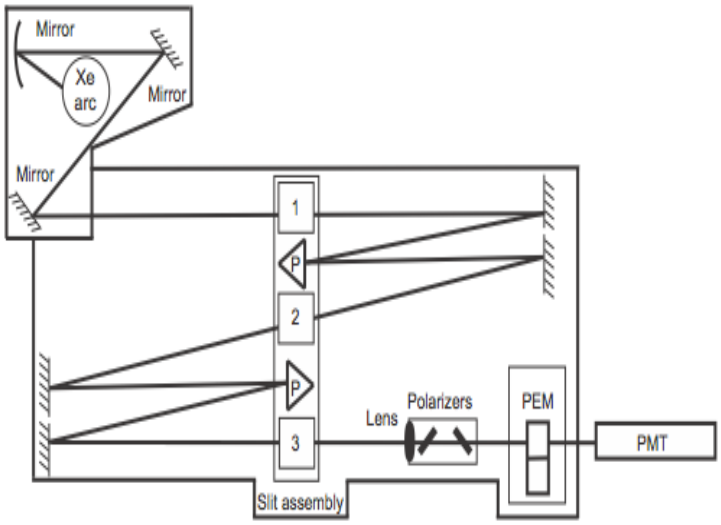


Figure 7: A diagram of the optics that go into making a CD spectropolarimeter. It contains a xenon arc lamp, mirrors, prisms, slits, polarisers, a lens, a PEM and a PMT . The xenon lamp emits white light with a maximum flux in the 300 nm to 400 nm region. Mirrors direct the light to the slits and prisms. Prisms select the relevant wavelength of light, polarisers perfect the linear polarisation before the PEM makes it circularly polarised. [5]

constant. The pH should be checked at the temperature the buffer will be used at; some buffers have high temperature coefficients.[53,54]

Buffers can affect the CD spectra when they are used for solutions of proteins or other chiral molecules. Every time a CD spectrum is taken of a chiral molecule, the base line should also be scanned. This is the CD spectrum without the chiral molecule present, i.e. just the buffer solution. This should then be subtracted from the CD spectrum for the molecule of interest to obtain the true spectrum. This should also be done to check the quality of the buffer, as some can cause the high tension, HT, voltage to rise to a level where the CD spectrum becomes too noisy to be useful. This is due to the buffer having a high absorbance in this wavelength range. The high tension voltage is the voltage applied to the PMT.

Absorbance in a CD spectrum due to the buffer solution can cause degradation of the signal, to the point that the signal to noise ratio becomes too low. Kelly *et al.*[53] found that absorbance is wavelength dependent, and more noise is caused at short wavelengths, particularly the 190 - 200 nm region. The absorbance is measured by the HT voltage. This is the voltage applied to the photomultiplier tube (PMT). This means the PMT has to work hard in the short wavelength region. For good CD spectra the HT should be at most 600 V, but this depends on the CD machine.[53]

For this reason synchrotron light sources with much greater light fluxes than xenon arc lamps are used to gather what is termed SRCD, or synchrotron radiation CD. With a synchrotron, useable CD spectra can currently be obtained down to approximately 160 nm. However, due to noise from most desktop CD machines, most CD-to-structure estimation algorithms do not use data below 190

nm.

Unwarranted interactions, like chelation (the formation or presence of bonds between separate binding sites on the same ligand and one central atom) of essential metal ions can also cause trouble with phosphate buffers among others.[53]

6.2.3 Algorithms for circular dichroism pattern recognition

With knowledge of where the CD signals come from, and how to prepare the buffer, the data analysis stage needs to be considered, as the CD spectra are not easily read by non-expert CD practitioners. Despite being able to tell which structures are present, even the experts cannot say exactly what proportions of structures are present. For this statistical or learning algorithms should be used.

There are various methodologies for CD pattern recognition to find secondary structures of proteins: CDSSTR, SELCON3, VARSLC, CONTIN, LINCOMB, MLR, CDNN, SOMCD, K2d, K2D2, K2D3, as well as our own: SSNN.[13,26–28,55–57] Here is a brief review of the methods used. SELCON3, CDSSTR and K2d were tested and reviewed in attached papers. The methodologies are based on multiple linear regression (MLR), singular value decomposition (VARSLC, SELCON, CDSSTR), ridge regression(CONTIN), convex constraint analysis (CCA), self-consistent method (SELCON), constrained least squares analysis (LINCOMB), and neural network (CDNN, SOMCD, K2d, K2D2, K2D3, SSNN).[58] The structures for these databases generally come from X-ray crystallography. See Figure 8.

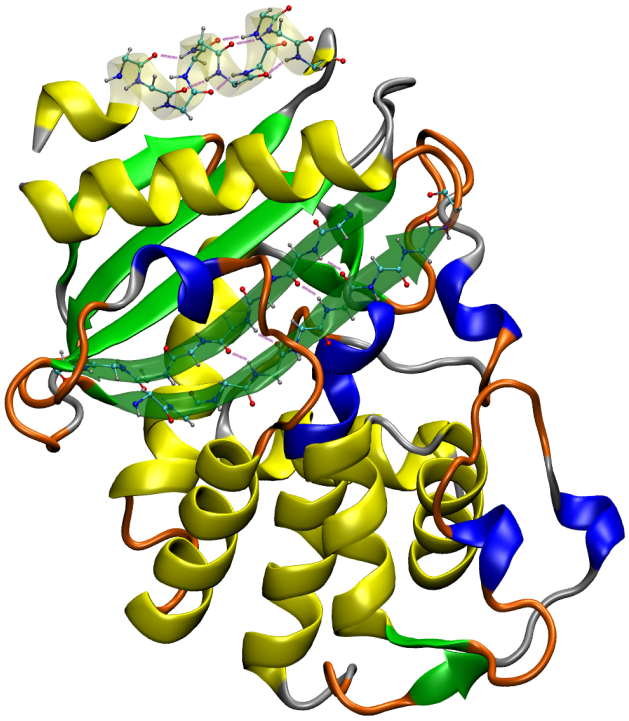


Figure 8: A picture of a protein with the different structures present: $\beta-$lactamase: $\alpha-$helix in yellow (long helices), 3-10 helix in blue (short helices), $\beta-$sheet in green, turns in orange, random coil in grey. The atomic-scale detail, along with hydrogen bonds, is also shown for two small sections: α-helix and β-sheet sections. The hydrogen bonds are the purple/pink lines between the red and white atoms. Atoms are carbon in turquoise, hydrogen in white, oxygen in red, and nitrogen in dark blue.

Greenfield's review

In reference[58] Greenfield compared the structure prediction qualities of MLR, LINCOMB, SVD, CCA, CONTIN, SELCON, Böhm *et al.*'s NN, and K2d.

Most of the algorithms are good at predicting the α-helix structures from CD, but have more difficulty with all other structures. The helix structures were predicted by all algorithms with Pearson correlation coefficients (P) of 0.88 or higher, and standard deviations (σ) of usually about 0.1. The β-sheet and β-turn structures were not as well done, and were far more variable. β-sheets were predicted with P of 0.00 to 0.91, and σ of 0.07 to 0.28. The β-turns were predicted with P values of -0.56 to 0.84, and standard deviations of 0.05 to 0.27.

K2d does not predict the structure type β-turns, it uses the structure types 1) α-helix, 2) β-sheet and 3) other.[26,58]
MLR (non-constrained least squares analysis) predicts α-helix, β-sheet parallel and anti-parallel, β-turns, and random conformations.

LINCOMB predicts α-helix, β-sheets, β-turns, and "random coils". Böhm *et al.*'s neural network predicted helix, parallel and anti-parallel sheet, turns, and remainder.

A likely reason these algorithms predict β-turns so poorly is because there are at least 4 different types of turns, which do not have very similar spectral features associated with them.

MLR predicts helix well, sheet with some correlation with experimental, and turns very poorly. Similar to MLR, LINCOMB predicts helix well, sheet with some correlation (very variable), and turns rather badly.

SVD predicts helical structures very well, but sheets and turns extremely variably, and rather poorly. It predicts as follows: sheets $0.00 \leq P \leq 0.68$, and

$0.12 \leq \sigma \leq 0.27$, and turns -0.56 $\leq$ P $\leq$ 0.22, and $0.1 \leq \sigma \leq 0.27$. Here P is Pearson's correlation coefficient, and σ is the standard deviation. So a large P, closer to 1.00, is better (higher correlation), and a small σ is good (small error).

CCA makes good predictions of helical structure, but its estimates of sheets and turns are worse than that of other methods.[58]

CONTIN gives good helix predictions, and its turns predictions are better than MLR, SVD and CCA. VARSLC produces excellent helical structure predictions, while sheet and turn predictions are much improved over the above, especially the sheet predictions. Data needs to be collected to 184 nm to make useful predictions/estimations, which is a handicap.[58–60]

SELCON's predictions for globular protein helix, sheet and turns are all very good. There does not seem to be much detriment to using a reduced data range of 240 nm - 200 nm. The 1996 version of SELCON worked well for globular proteins, but did not predict well the structures of polypeptide with large percentages of sheet; it over-estimated helix content, while under-estimating sheet content quite badly. For a comparison between SELCON3, a more recent version of SELCON, and SSNN see Paper 1[17] and Paper 3[4].

The Neural Network for protein secondary structure estimation that was written by Böhm *et al.* was extremely good at predicting helical, and antiparallel sheet structures. The correlation coefficients were 1.0 and 0.91. The only problem appears to be when the wavelength is restricted to 250 nm - 200 nm, the sheet estimation is negatively correlated.[61]

K2d gives good sheet estimates when the wavelength range is restricted. It does not estimate turns.

For these applications, most of the algorithms are operating in the range 240

nm - 200 nm, VARSLC uses the range 260 nm - 184 nm, and SELCON uses 260 nm - 200 nm. Böhm's NN (neural network) used 83 element input vectors, representing the intensities at 260 nm - 178 nm.[58] MLR does not require an exact value of the protein concentration. According to Greenfield, CCA does not find the secondary structures of unknown proteins easily, given no extra information, but it is very good for analysis of spectra of proteins and peptides with regard to temperature, pH and ligand binding.

MLR, SVD and CCA do not have selection procedures for known spectra to make model spectra of unknown proteins. The known spectra do not necessarily have similar features to the test spectra. This leads to the models being very dependent on the spectra which were chosen to make the models. So, the suggestion was to introduce these similarity selection measures. That is what has been done with (1) ridge regression, (2) variable selection, and (3) neural networks; examples of these are (1) CONTIN, (2) VARSLC (which is SVD with variable selection) and SELCON (VARLSC modified), and (3) K2d. Algorithms with selection perform better than those without. SELCON, or self-consistent method by Sreerama and Woody gains a speed advantage by routinely removing the spectra least like the test spectrum.[58,62–64]

It should be noted that the algorithms that produce the best spectral predictions do not necessarily produce the best structure estimates. Greenfield points to CONTIN as "almost always [giving] excellent agreement between the experimental and calculated CD curves, even when the fits are relatively poor compared to other methods." Greenfield says that K2D often produces very poor spectral models, while making very good structure predictions.[58] Another example of this is CDSSTR; it produces beautiful spectral models, among the best of all CD-to-

protein-structure algorithms, but its structure predictions/estimations are not as good as some competitors (e.g. SELCON3 and SSNN). SELCON3 has possibly the best structure predictions besides SSNN.[58]

The most sensible way to analyse CD spectra to get structure predictions is to use at least a few different methods, and see where they agree. If the predictions vary greatly, then one cannot have great confidence about the structures present.[4,58]

Greenfield says that if one needs structure predictions, but does not know the concentration accurately, then non-constrained least-squares analysis programs like MLR are the only options, but these do not give the best predictions. This issue is dealt with in reference[38] also known as Paper 2 in this thesis.

K2d

In their 1993 paper, Andrade *et al.* introduce K2d, their SOM methodology for estimating protein secondary structures from CD spectra.[26] This paper lends more detail about K2d and its proteinotopic map, than given by Greenfield, mentioned above. A proteinotopic map is a topological map of some information about proteins.

The training set for the K2d SOM is 24 CD spectra. One of these proteins is poly-L-lysine, which has different CD spectra depending on pH and temperature; for this reason it is used as a model protein for the various spectral features. Eighteen of the proteins have known (static) structures, and 3 are constructed from 15 proteins of known structure originating from Chang *et al.* 1978.[65] More information on this reference set can be found in Yang *et al.* 1986.[66]

The K2d map is a 13x13 square lattice of nodes and, like other SOMs, it interpolates between experimental CD spectra, but does not extrapolate. For this reason, unlike some methodologies, K2d does not give structure estimates with negative values, which would be physically impossible.

As Unneberg *et al.*[13] point out, the database size determines the possible sizes of the map: a map needs to have enough data space to store all the examples, but must not be so large as to make all BMUs in a region essentially the same, and negate the use of multiple BMUs. That would cause the SOM to produce poor spectra models. So with 24 spectra in its training set, K2d needs a smaller map than a SOM with more spectra would.

Due to the CD data used by K2d being of the wavelength range 240 - 200 nm, there are 41 data points. The SOM does not use CD spectral data beyond 240 nm, as there is not much information from the peptide backbone in that region. For reasons of wavelength range and how much information can be extracted from such a range, there are 3 structures types estimated by K2d, α-helix, β-sheet and "random coil", or other. There are not enough electronics transitions that emit light in this region to provide information to resolve more than α-helix and β-sheet. The n $\rightarrow \pi^*$ transitions are between 190 nm and 200 nm. This means that there are only 2 independent variables, for the helix and sheet; the random coil value is calculated by subtracting the helix and sheet values from 1.00.[26] It should be pointed out that random coil is a particular structure type, though rather difficult to generate, but is often used as a category for everything that does not fit into any other categories, like the group "other" structures.

Conversely there is a lot of information from the CD spectrum with regard

to the structure of the protein, or other chiral molecule, in the short-wavelength range below 200 nm. However, this data is much harder to access due to the high energies required, and most desktop CD spectrophotometers cannot get CD spectra in this range without increasing the absorbance to such levels that there is a great deal of noise (at the time of writing, May 2014).

As mentioned above, this is a reason to use SRCD, the synchrotron overcomes this absorbance issue, see the section on SRCD, section 6.2.4.

Some later algorithms inspired by K2d do use data in the 200 - 190 nm range. The Andrade team point out that while collecting spectra without the shorter wavelength region, the methodology cannot hope to reliably estimate multiple β structures, hence why they decided to estimate just one helix structure and one sheet structure.

K2d uses a parameter to track the training progression of the SOM, it is called the distortion parameter, D. D is the sum of all the distances between each training set spectrum and the model spectrum most similar to it. If this value is generally decreasing, then training is headed in the right direction. If it stalls, and does not continue decreasing, then training has stopped being useful.

With regard to the success of clustering CD spectra, the K2d SOM team found that each of the three structure labels claimed a corner of the map; in the random coil corner, those spectra of proteins with high random coil clustered, with the same result for sheet- and helix-rich proteins.

K2d uses a BMU count of 2. This is the number of BMUs, or model spectra from the map that go into making the final output model of the test spectrum, see above. Studies of the effects of using different map sizes revealed that a

smaller number of BMUs constructing the model spectrum are appropriate for smaller maps. The size of the map for the SOM we made (SSNN) is 40x40 nodes, and K2d's map is 13x13 nodes. In our work we found that SSNN made better structure estimations when using 3 BMUs for a map size of 20x20, so this is in agreement with the K2d team findings.

Unneberg *et al.*[13] note that K2d maps were the product of the averaging of several map-trainings; SOMCD (see below) and SSNN only require one training.

SOMCD

SOMCD is a SOM methodology inspired by K2d, and developed by some of the same team that made K2d.[13] In addition to the functions of K2d, SOMCD also estimates percentages of β-turn structures, the training set spectra now number 45. The resulting estimates of the different structure types are more uniform than from K2d: the estimates for different structure types are about as good as each other.

SOMCD uses 45 spectra, an additional 21 from that of K2d. SOMCD was tested with leave-one-out cross-validation where the SOM is trained with all but one training spectrum, and the one left out serves as the test spectrum, then this is iterated until each spectrum has acted as the test example, and all others the training examples. The method was used to validate training parameters like map size and initial neighbourhood radius, among others, for SOMCD as for SSNN. A map size of 16x16 was chosen for SOMCD based on these validations.

SOMCD uses three BMUs, an additional one compared with K2d. This makes sense given the larger map and training set. One BMU would produce just a

model of one of the training set CD spectra, so a few are needed to take elements from different training set spectra to make a model of the test spectrum. There need to be more than one BMU because just one would produce a model of the training set spectrum at those coordinates. If there are two or more, they can come together to make new and different models of spectra not on the map.
The additional structure, β-turns, seems to have been harder to fit in with the clustering, as the map at the turns level does not show an obvious peak, while each of the other structures does have a peak in its corner. Indeed the team state that this is the worst predicted structure. It is interesting to note that the team use a histogram to show that the mode of turn structures in the training set is 10-15 % turn per protein. This is a reason for the poor estimation of turn structures; only one protein in the set has more than 20 % turns.

The SOMCD group use the som_pak or SOM_PAK[67] package, made by Kohonen *et al.*, to study the continuity of weights in the map. They wanted to find that neighbouring nodes hold very similar vectors (model spectra). A figure shows that this is the case for almost all regions of the map, see Figure 9. In this way som_*pak* helps the user to see if the SOM has clustered well.

The SOMCD Discussion section shows that the structure predictions for all similar structure types were better estimated by SOMCD than by K2d; this is based on the RMSD and the Pearson correlation coefficients. There is also a web version of SOMCD for testing one's CD spectra.

K2D2

Another K2d-inspired algorithm for protein secondary structure estimation from CD spectra is K2D2.[28] This was developed by Perez-Iratxeta, and Miguel Andrade-Navarro who was from the K2d team. They also set up a webserver

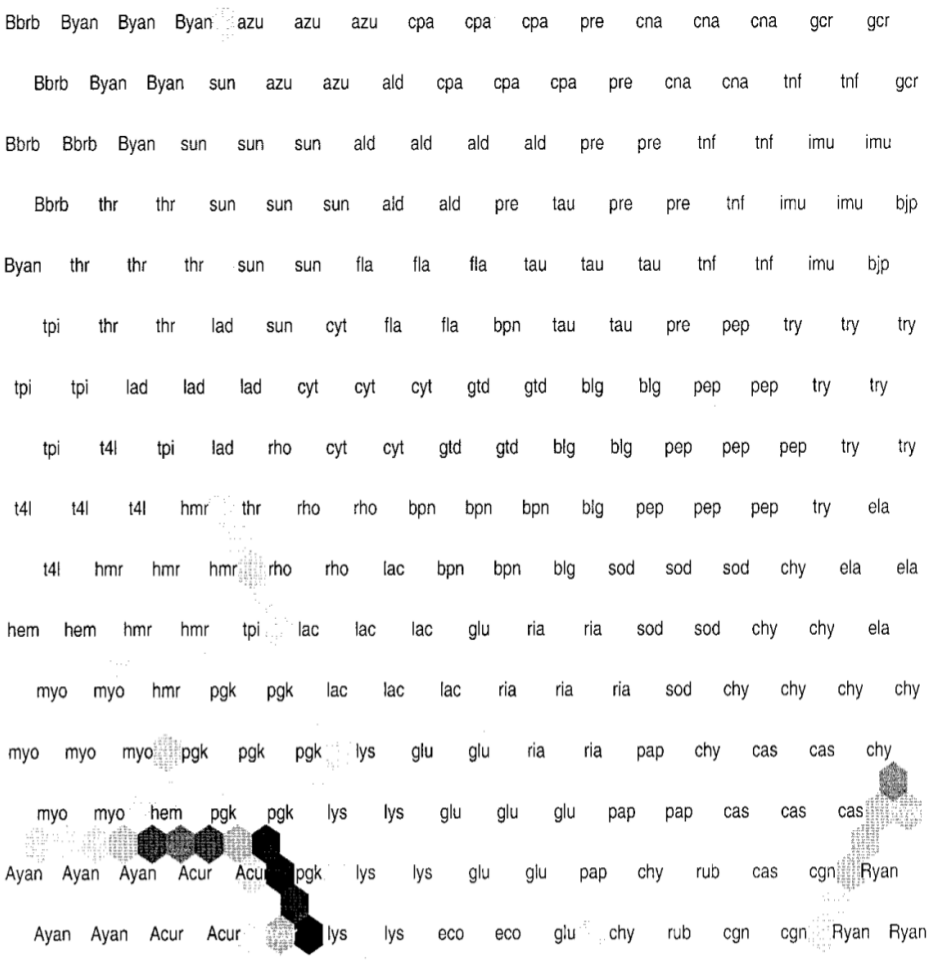


Figure 9: Figure from SOMCD paper showing that like proteins are clustered close together. The grey scales represent the Euclidean distances between nodes: darker grey means the nodes are further apart.

(http://k2d2.ogic.ca/) to handle protein structure enquiries to their SOM.

This methodology only uses helix and sheet structures, like K2d does, but they have extended the wavelength range to 190 nm, and increased the reference set protein spectra to 49, or 43 for validation, both improvements help to improve the structure estimations over K2d. The average RMSD values for helix and sheet prediction were 0.08, and 0.09. These contrast with K2ds 0.11 and 0.14 averages.

The K2D2 group also made use of the SOM_PAK package, and they changed the map size to 18x18 nodes. Like K2d, the final map was the average of many maps; in this case 100.

Noting that CD spectra-to-structure-estimation software packages do not currently use membrane protein information, they included in the training set 13 transmembrane (TM) proteins. TM proteins are usually extended between the extracellular space, across the cell membrane, to the intracellular space, so from just outside the cell into the aqueous volume inside of it, see Figure 10.

Unfortunately, for the team, and CD practitioners, this resulted in a decrease in performance. They conclude that TM proteins would need a separate map trained with only TM protein spectra, to get good estimations of structure. We concluded the same, but have not made a trained map with TM proteins. We did, however, test TM proteins (from another laboratory in the department) with our globular-protein-trained SOM; the results were predictably very poor. Maps trained on globular protein spectra should not be used to estimate the structure content of transmembrane proteins; that is the conclusion from that paper, and from the author's experience.

The K2D2 group used "more than 6" BMUs to make the model spectra of the test proteins, a relatively large number compared with K2d, SOMCD and SSNN. K2D2 is platform independent and was written in Perl programming language.

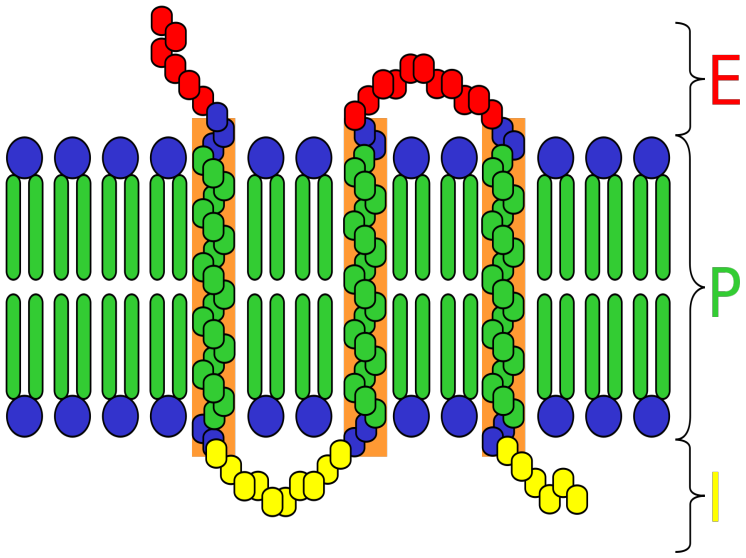


Figure 10: A simple 2 dimensional representation of a transmembrane protein, E is extracellular space (outside the cell, which is an aqueous environment), P is the plasma membrane (inside the membrane, which is a non-polar, phospholipid environment, so it's oily), I is the intracellular space (inside aqueous environment of the cell).[7]

K2D3

In a further paper written four years later, in 2012, the two authors from the K2D2 group with the addition of Caroline Louis-Jeune wrote a follow-on K2d-inspired code called K2D3[27]. Here the team sought to greatly increase the training set by including theoretical spectra generated by DichroCalc by Bullheller and Hirst[68]. The aim was to make a non-redundant collection of protein spectra that would represent most of the proteins in the Protein Data Bank, PDB, `http://www.rcsb.org/pdb/home/home.do`[69]. They report improvements over K2D2, especially for sheet predictions, and in the range 240 - 200 nm. Indeed, their results show that there is an improvement for that range, but for the 240 - 190 nm range the results were not statistically significantly different.

To compile a training set, Louis-Jeune *et al.* searched the PDB, and found 140,624 protein chains, then clustered them into 23,406 groups, from 8,265 PDB files (multiple protein chains per file). For each group they used selection criteria borrowed from NCBI's Structure Group[70] that aim for proteins with most similarity along their full sequence. More criteria were applied until there was only one protein in the group. The PDB file for *that* protein represented the group. With these structures DichroCalc was used to calculate spectra for 16,050 protein chains.

When it came to validation of the methodology, the inclusion of proteins with different sequence lengths in the SOM caused the additional problem of the BMUs of a protein having very different lengths (different numbers of residues making up the proteins). To manage this, the group introduced an additional selection criterion of having a similar length to the query protein. This produced slightly better correlations for α-helix predictions, and much better for beta-sheet pre-

dictions.

K2D3 group tested these theoretical spectra to see if they had correctly reproduced known protein spectra using a benchmark dataset of 83 protein spectra. They report that the models matched the experimental spectra well, and noticed very good matching in the 210 - 240 nm range. Although most information about structures present is in the short wavelength region.

The plot of Pearsons correlation coefficient against wavelength shows that most of the average spectrum correlates with r = 0.8 approximately, but at 200 nm this drops precipitously to about 0.2. The correlation is between the model and the experimental spectrum, see Figure 11.

The K2D3 team assigned secondary structure compositions to experimental and theoretical spectra using Define Secondary Structure of Proteins, DSSP; they only used helix and sheet types.[1]. K2D3 was subject to leave-one-out cross-validation with the full set of 83 proteins in the BENCH83 dataset to validate all of the parameters for training.

K2D3 has reduced performance accuracy when dealing with very high beta-sheet proteins, due to its reference set not covering these structures. When predicting structures of polypeptides with nearly 100 % sheet structure K2D3 predicts sheet percentages far below their actual values, and helix percentages far above. A polypeptide they mention, with regard to this, is poly(L-lysine). This is surprising, as we have studied this peptide in one of our papers using our SOM, SSNN.[4] We managed to get the low error of 0.032 (NRMSD: normalised root mean squared deviation, see equation 3) from our spectral model of polylysine with very beta-sheet structure.[38]

This K2D3 application is also available on a webserver, `http://k2d3.ogic.`

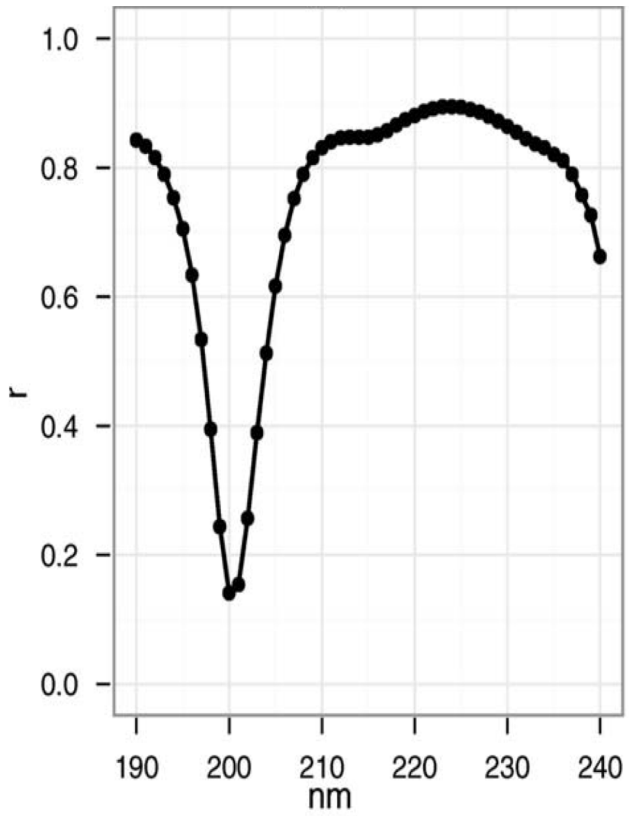


Figure 11: Plot from the K2D3 paper by Louis-Jeune *et al.* 2012 showing the Pearson correlation coefficient against wavelength predicted by DichroCalc.

ca/. The server accepts differential absorption units ($\Delta\epsilon$) or mean residue ellipticity units (θ). The size of the protein may also be entered, which, as said above, might greatly help the beta-sheet prediction. The size of the protein can be reported in number of amino acid residues or in molecular weight, kDa. They report that many of the queries of protein structure to their K2D2 server have the range 240 - 200 nm.

Louis-Jeune *et al.*[27] say that alpha-helix predictions do not leave much room for improvement, as they are so accurate, however they saw an improvement in beta-sheet prediction from using K2D3.

6.2.4 Synchrotron radiation CD

Due to HT voltage problems with CD spectra caused by buffers and samples absorbing too much of the circularly polarised light, blocking it from interacting with the sample molecule, and being registered by the PMT, synchrotron radiation is used to produce much better CD data, and at wavelengths down to 170 nm, as reported in Kelly *et al.*[53]

Wallace *et al.*[71] use SRCD because it only needs small volumes of solution, it is high-throughput, it uses very short wavelength, high intensity, UV light. More electronic transitions[72] are visible in the short wavelengths below the 190 nm used by most benchtop CD machines in most CD laboratories. Two examples of peaks and troughs in the vacuum ultraviolet (VUV) region are the 160 nm and the 175 nm transitions that are thought to originate from interamide charge transfer transitions.[73]

The first use of SRCD was in 1980, but only at the turn of the 21^{st} century did anyone manage to obtain high-enough quality SRCD spectra to be able to analyse the secondary structures of proteins in solution at wavelengths as short as 160 nm.

As of the writing of that article, there were only 3 sites where SRCD spectra could be gathered, with another in construction, so SRCD is not nearly as widely available as standard CD.[71] The sites were Synchrotron Radiation Source (SRS) in the UK, Aarhus Storage Ring (ASTRID) in Denmark, the National Synchrotron Light Source (NSLS) in the USA, and there was another light source under construction: the Beijing Synchrotron Radiation Facility (BSRF) in China. Nevertheless, SRCD provides some great benefits, like a much cleaner signal due to higher signal-to-noise ratios than conventional CD laboratory machines. This reduces the volume requirement of the protein in solution. Other benefits are: the chance to work with CD samples that have high concentrations of absorbing materials such as buffers and salts, and shorter measurement times.

Indeed, compared with crystallographic work, the SRCD sample volumes are 100^{th} to 1000^{th} the size, and the measurement times are 1000^{th} to $10{,}000^{th}$ the length.

The possibility of gathering CD spectra to 160 nm means that sheet and helix peaks can be more easily resolved, as their amplitudes at wavelengths shorter than 190 nm have opposite signals, while the sheet component can be drowned by the helix component at wavelengths slightly longer than 190 nm. This means that, with SRCD spectra, the sheet estimations can be more accurate than with spectra from regular CD laboratories. Part of the reason for these improvements is due to the higher number of structural types that can be resolved due to the use of shorter wavelength data.

For these reasons, SRCD could become a valuable method for high-throughput functional and structural genomics, and proteomics screening investigations.

From at least the time of writing that paper (2001) the Wallace Group saw that determining membrane protein secondary structures would be a sensible

expansion of their work, and they have since done so.[74] The database is now available on the Dichroweb site[75]. Membrane proteins are such an important expansion of their work because they comprise about 60 % of drug targets and approximately a third of the proteins in the human body.

6.2.5 Crystal structures

In order to have CD spectra with known structures (or conformations) for reference sets, work needs to be done to find those structures, and X-ray crystallography is one way to get them. We attach these structures to the CD spectra of known proteins to form our training sets. First the proteins have to be crystallised, exposed to X-rays, then have their patterns analysed to calculate their structures.

Research teams around the world have managed to crystallise many thousands of proteins, and determined their structures, but not all proteins take well to the crystallisation processes used.[13] However, there is another problem: proteins labelled with structure types in different ways. Searching for a protein database with proteins all labeled in the same way, thousands could not be found, in fact not even hundreds. The search only managed to reveal 48 similarly labelled proteins: (CDDATA.48 from CDPro), which we used to train SSNN.[56,62]

Our group later added five more spectra and structures: 2 extrapolated to have 100 % α-helix, and 3 100 % random coil, which brought the count to 53 ("CDDATA.48+5"). The last 5 spectra are Sulf–KK (100 % Random coil), 100 % α-helix calculated (extrapolated) from Myoglobin, then Estimated 100% α-helix protein from Aurein 2.5 peptide, then N–formyl acetic acid (100% Random coil) and N–acetyl valine (100% Random coil). Every group labels their protein sec-

ondary structures in slightly different ways, this is due to personal interpretation. Some have 5 structure types, some have 6, different groups determine in different ways what is a helix, what is a sheet, and what the distorted versions of those spectra are. Some have poly-proline II helices, some don't; there are numerous variations.

6.3 Myoelectric signals for control of robotic upper limbs

Alongside the work on pattern recognition of CD spectra and their structures, the project also looked at controlling robotic limbs. It was realised that the ML SOM methodology could be applied to other datasets, and an opportunity arose to work on myoelectric signals; the electrical information from muscle contractions; for the control of prosthetic robotic arms. The myoelectric signals are extremely complex for a human to recognise, and program into a control system, so a machine learning technique was applied.

What follows is some background on the myoelectric signals, and their use to control robotic limbs.

Myoelectric signals (MES) are the electrical signals produced by impulses from the brain, and they can be measured from muscles when they move. Research into control of robotic limbs using myoelectric data began in the 1960s, but was first theorised in the 1940s. However, the technology did not have a large impact clinically until the 1970s. This was made possible by the reduction in cost, size and energy requirements of semiconductor technology.[76]

A reported 12,000 people had upper-limb abnormalities in the UK in 2001. Globally this number has been reported to be as high as 3 million (of the 10 million amputees).[77] According to the NHS[78], about 5000 to 6000 major limb

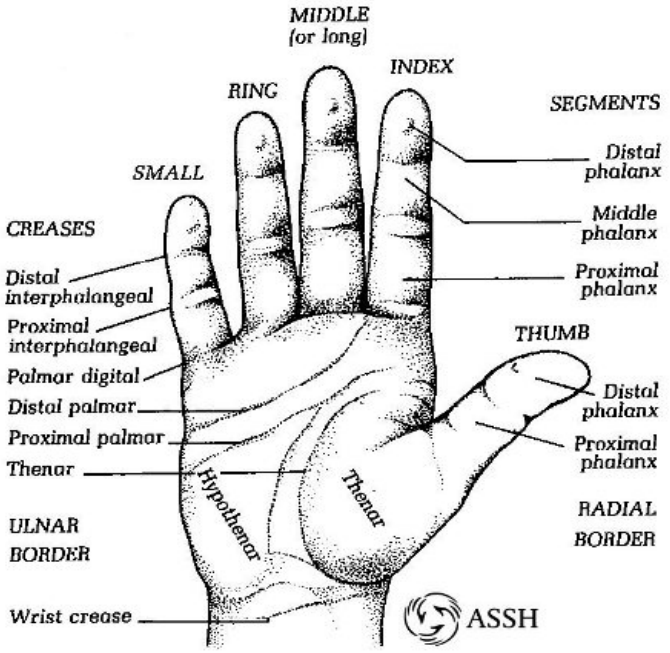


Figure 12: An annotated diagram of the intact human hand, by the American Society for Surgery of the Hand.[8]

amputations are performed every year in the UK. 70 % of the cases are due to loss of blood supply, this is called critical ischaemia. 57 % of upper limb amputations are due to trauma, and diabetes sufferers are 15 times more likely to need amputations than the general populace. The most common age group to receive amputations are the over 70s, and men have twice the risk of needing one than do women. The CDC in the USA reports that in the US approximately 1,450 people are born each year with upper limb deformities, only about 700 with lower limb deformities.[79]

When making a prosthetic to replace the hand and arm, it is good to have a model of the existing biological system that evolved to be so useful in our daily lives. Figure 12 shows the names of sections of the human hand for reference

when describing parts of it in the discussion below.

The human hand is underactuated, meaning that there are more degrees of freedom for the hand than there are actuators, or from a ML point of view: there are more outputs than inputs.
The distal phalanx of each finger (shortest segment furthest from the wrist) cannot move independently, but can be bent in towards the palm when the middle and proximal phalanges are bent towards the palm. This type of movement enables the hand to grip an object of any shape without having special movements for each shape. Said in another way, the hand has many degrees of freedom.

The human hand movements require a great deal of coordination between different muscles in the hand, wrist and arm, and the mastery of this may take several years. There are various levels of control present, from the low-levels for fine-tuning the forces required to maintain grasp by individual digits all the way up to the high-level which decides the type of grasp needed.

Those in the field of robotic prosthetic hands try to mimic the functions and strengths of the human hand, so there are some pre-set movements or grasps that are used. The prosthetics community suggests there are five basic movements that the human hand regularly uses:

1. The pincher grasp (with thumb, index and middle fingers together)
2. Key grasp (thumb resting on the side of the index finger)
3. Hook grasp (used to carry things e.g. books, suitcases)
4. Spherical grasp (for holding a ball)

5. Cylindrical grasp (for holding a cylindrical object)[9,80]

The work reported on in this section of the thesis is to do with the software platform for myoelectric robotic limb control called BioPatRec developed by Max J. Ortiz-Catalan *et al.* at Chalmers University, and Integrum both in Sweden.[81,82] The BioPatRec control system is designed for use with a variety of robotic wrists and hands with 4 fingers and one thumb each, with the 10 pre-set movements plus rest:

1. open hand
2. close hand
3. flex hand (move hand from wrist in direction of palm)
4. extend hand (move hand from wrist in direction of back of hand)
5. pronation (place palm down)
6. supination (place palm up)
7. side grip or hook grasp (e.g. for opening a fridge door/carrying suitcases)
8. fine grip (similar to pincher)
9. agree (thumb up, fingers bent in)
10. pointer (pointing with index finger)
11. rest (resting position of hand)

The average human hand weights about 500 g, so the challenge includes making a hand that has enough actuators and possible movements and positions, while weighing no more than 500 g, and fitting onto a person without looking incongruous (e.g. too large). If the aim is to make a small enough prosthetic

for anyone, the researchers should concentrate on the smallest arms and hands.[83] Variety Village even produce prosthetic hands, wrists and arms for children of different ages.[84,85] In the past, for certain prosthetics, this was not possible for anyone but adult males, due to the combined size of all the components.[84] Gow *et al.* decided to design their prostheses for the smallest and shortest people so that everyone could have something tailored for them, by using a modular system.

Of course not all people with limb abnormalities have the same issues, there will be those with partial loss of hands, some will need wrists, some elbows, and some will need everything up to and including shoulders. Of course there need to be available both left and right hands and arms. A robotic prosthetic system needs to be customisable for each person.

According to Ajiboye *et al.* in[86] surface electromyogram, or sEMG is the state-of-the-art hardware for externally-powered transradial prostheses. A transradial prosthesis is for the arm below the elbow. However, most of these are two site devices with one DoF, and some have two DoF, but to use one component, the other has to be switched off. To attain full multifunctional myoelectric control, the system must allow all functions to operate simultaneously with other control functions, and not inhibit them.

In[86], Ajiboye *et al.* used 4 states of activity or input membership functions: High, Medium, Low and Off. They used a threshold for the dividing line between rest state and any other state. This was set to the mean value recorded from subjects when they were not attempting to move their arms. The mean for performing some movement was called MED. The LOW and HIGH were chosen to so that there was a 50 % overlap between two neighbour membership functions.

To collect the signals, the intact limb test subjects underwent the following movements: wrist extension, wrist flexion, ulnar deviation, and finger flexion repeated eight times.

The control algorithm used by Ajiboye *et al.* is a fuzzy logic method. The system is made up of three elements: 1) the fuzzifying stage: numerical inputs are converted to linguistic variables using input membership function; 2) a pattern classification by processing linguistic inputs; 3) the defuzzifying stage outputs a single number from the inference rule base linguistic outputs. The core part of the fuzzy system is the inference rule base (IRB). This carries out the classification by pattern recognition of the MES signals, and works by studying the relative amplitude of the EMG signals. The IRB is a set of IF (condition based on signals present), THEN (outcome movement) statements. The rules can be given differing weights depending on scenario likelihoods.

In the real-time tests the subject performed a movement, then returned their arm to a fully rested state, as is standard with these prosthetic robotic limb tests. Although, this system can recognise desired movements and transition between them without returning to rest between them, which was also tested. Their real-time results for all 7 movements achieved accuracies ranging from 94.79 % to 98.27 % for the 4 subjects tested. The seven movements were: off, wrist extension, wrist flexion, supination, pronation, ulnar deviation, and finger flexion. Flexion is moving inwards (towards the palm), extension is moving outwards. Supination is placing the palm up, while pronation is palm down. Ulnar deviation or flexion is moving the wrist in the direction of the little finger.

The system developed by Schwartz *et al.* was tested with PVA, (population vector algorithm), optimal linear filters, maximum likelihood estimation, SOFMs, and a recurrent neural network. With this system, control of the prosthesis was performed by populations of single- and multi-unit motor cortex spiking activity. The real-time population vector is populated by the preferred directions of the units in the population that is recorded, put into a vector sum. The sum is weighted by the instantaneous firing rates of the units.

Schwartz *et al.* also tested optimal linear filters, which is essentially a sum of firing rates for all cells going backwards in time; it makes use of temporal information, unlike PVA. The maximum likelihood estimate is the movement direction that maximises the probability of observing a particular firing rate of a cell. Therefore, the discharge or firing rates of neurons determine the movement desired.[87]

The recurrent neural networks output vector managed to cluster after several iterations. The output vectors formed a clear peak, with one mode, in the region of an output movement. The peak showed which direction is most likely. Gaussian noise, from the input is removed by the nonlinear activation function in the output.

Another thing that Schwartz *et al.* comment on is that correlation within and between parameters has not been fully looked into with regard to decoding algorithms. It is well documented that various parameters involved in movement covary, this includes both intrinsic and extrinsic parameters of the arm.[88–90]

6.3.1 Time windows for MES

In pattern recognition of MES the recordings are usually divided into small sections called time windows, so that features can be extracted from them. If one were to find the mean of a MES spectrum over the entire recording, the data would be lost, one would get a flat signal. So the means of these time windows are used, along with other features specific to the methodology used. These may include zero-crossing rate, integral absolute value, variance, waveform length, autoregressive coefficients, Fourier transform coefficients, linear cepstrum coefficients and adaptive cepstrum vectors, and principle components from PCA. Autoregressive models model stochastic processes by assuming that future values can be predicted by past values[91,92].

Cepstrum results from taking the inverse Fourier Transform of a logarithm of the estimated spectrum. "Ceps" is "spec" backwards.[76,93–95] Engelhart and Hudgins in[76] used a majority voting to post-process the decisions made by their LDA classifier. For a decision point, the majority vote done for this sample was voted on by: 1) the sample, 2) m samples before it, and 3) m samples after it (2m+1). The winner is the class with the most occurrences in the (2m+1) voting window. Voting has the effect of removing spurious decisions.

Using a long time window for analysis (e.g. 256 ms) means that classifications or predictions will be more accurate, and there will be lower variance. However, longer analysis windows require more processing time, threatening control lag on the limb (time delay). If using a very short time window for analysis (e.g. 32 ms), then the error for each windows decisions will be higher due to higher variance. However, the number of decisions on the desired motion of the prosthetic limb will be higher too, this means the motion can be corrected more quickly, and

have more fine control. Also, the large number of votes makes averaging out the errors easier.

6.3.2 Other ML and MES advice

Before they can be studied, the MES have to be gathered from people's arm muscles, this is usually done with electrodes. Historic systems attempted to use a large number of MES recording channels (e.g. electrodes), but the computation was too much for the systems to handle. Later systems have used feature extraction to maximise the ratio of possible movements to MES recording channels or sites, for example Hudgins *et al.*[96] in 1993.

Most algorithms require that various sources of variance be explicitly specified, as some optimal function is being modelled to the cell response. However, the SOFM approach is partially immune to this failing due to the fact that it clusters its output nodes based only on the similarities of its input patterns. PVA also has some resistance here, due to its ability to deal with two parameters (or maybe more) simultaneously: direction and speed of the arm.[87]

There are two nearly universal factors that determine the success of all the algorithms; uniform parameter distribution and unimodal tuning functions. The parameters here account for variance in the spike train. Clustering parameters in a non-uniform way leads to more damaging noise. A unimodal tuning function is useful because, even in Machine Learning techniques, where tuning functions are likely to have many modes, this multimodal nature causes a decrease in the likelihood of locating an obvious winner or BMU.

6.4 Contribution to knowledge of the research reported in this thesis

6.4.1 CD spectra fitting for protein structures

For this project the following has been developed: SSNN, a SOM artificial neural network software methodology for protein secondary structure estimation from circular dichroism spectroscopy, that compares well with methodologies that have similar aims. The tests performed showed that SSNN, the author's methodology, has produced better estimates of structure than all competitors. When compared based on individual structure types, SSNN performed better on most, and on others equalled the best for that structure type.[4,17,38]

6.4.2 Concentration correction

To help ensure the structure predictions are accurate, and all given knowledge is correct, SSNN was adapted for finding the correct concentrations of globular proteins in solution. SSNN was tested by running multiple structure estimation tests from CD spectra of proteins with unknown, unavailable structures. The best spectral models showed the region of the most likely correct concentration given as a number relative to the user-reported concentration. The best spectra models were selected based on their NRMSD values. For this reason two versions of SSNN were produced; one to run SSNN just once for structure estimation, and another to search for the correct concentration and make structure estimates at each concentration step.

For example, if there was a protein in a buffer at a reported concentration of 0.1 mg/ml, and the CD spectrum was run through SSNN, the CD spectrum would be multiplied by 0.1, 0.2, 0.3, etc. up to perhaps as high as 4.0. If the

NRMSD of the model spectrum with respect to the original CD spectrum was found to reduce between 1.0 and 2.0, then increase again, and the 2.0 was the smallest point, then the original concentration would be believed to be half what it was reported: 0.05 mg/ml. It would need multiplication up to twice the original to become a spectrum that is most recognised by the SSNN methodology, and therefore most likely to be realistic. Thus the conclusion that it is in fact only half of the reported 0.1 mg/ml.

Each of these versions is *also* in two divisions: one to just test the structures or concentrations of unknown proteins, and another to re-train SSNN with new data sets. So this methodology can be applied to any dataset; the data size can be changed and various training parameters can also be changed for optimisation. The team wanted to make this useful software application family available to everyone, so all these above versions of SSNN were made downloadable by the public on the Rodger Group[97] website along with detailed guides on how to use both. There are Windows- and Mac-compatible stand-alone MATLAB applications of these.

6.4.3 HASSANN for BioPatRec

Continuing the biomedical engineering vein from the insoles research, in the final section of the PhD project SSNN was adapted for use as part of the BioPatRec platform, so it became known as "HASSANN".

BioPatRec is a modular software platform written in MATLAB, that takes MES data, and uses them to enable people to control robotic prosthetic arms.[98] To gather data for BioPatRec, electrodes were placed on the upper arms of up

to 20 test subjects (different numbers for different tests), some with incomplete arms, some with full, healthy arms. Four electrodes were used, producing four channels of myoelectric spectra.

We used the BioPatRec_TVA version of the platform. There are five stages to using BioPatRec for control of robotic arms:

1. Data collection
2. Signal processing
3. Feature selection and extraction
4. Pattern recognition
5. Real-time control (movements are sequential)

Once the real-time sequential work is done, a later stage of the development can be looked into: real-time simultaneous control, here a few movements can be performed at once.

The BioPatRec development team have tested several other different pattern recognition algorithms. For this PhD the author has developed and tested HASSANN, or Hand Activation Signals SOM Artificial Neural Network, to complement the BioPatRec research by other machine learning experts. Other algorithms tested by the BioPatRec team include Regulatory Feedback Networks (RFN), Linear Discriminant Analysis (LDA), and Multi-layer Perceptron (MLP). BioPatRec is hosted online at http://code.google.com/p/biopatrec, where there is also a wiki on its development and updates.

HASSANN is derived from SSNN, Secondary Structure Neural Network. There are 11 movement categories: 10 movements, and rest. A rest category is important so that the limb only moves when the person controlling it desires it. It needs to have a threshold level of electrical activity, as there is electrical activity even when the controller does not intend to perform any tasks with the limb.[76,98]

6.5 Further work

6.5.1 Diabetes patient insoles and clustering academics for collaborations

As SSNN/HASSANN can be applied to any dataset, to solve many different problems, we wanted to use it for problems we had. SSNN has also been applied to such diverse goals as helping to select the most comfortable, least harmful insoles for the shoes of diabetes sufferers, and suggesting collaborations between chemistry academics (both not shown). The work on selecting insoles for diabetes sufferers was done for an Engineering masters student as a side-project to this PhD, and the collaborations between academics study was done with the soon-to-be-head of the Chemistry Department, Alison Rodger. Both side-projects are unpublished work, although the insole selection work is reported on in the masters thesis of Okem Molokwu, supervised by Evor Hines, then of the University of Warwick School of Engineering.

When people who have diabetes move into advanced stages of the disease they may lose sensitivity in their extremities, such as their feet. This can cause them to walk with gaits damaging to their feet. For this reason Molokwu's masters research focussed on looking for ways to select insoles that caused the least damage to the patient's feet.[99] This was done by recording the forces present at many

locations in the shoes of individual diabetes patients while they walked with different insoles, then applying machine learning or "intelligent systems" methods to the data. SSNN was one of the methods used; it made some good predictions, but could benefit from some optimisation.

For the clustering of Chemistry academics study, the SOM was trained on the research collaborations of 20 professors, and tested on research collaborations of the 24 non-professorial principal investigators in the department. The academics were listed in alphabetical order and a 44x44 matrix created. Academics were asked to indicate with whom they collaborated. In their row of the matrix each collaboration was indicated by a 1, while a 0 denoted no collaboration. Intriguingly the matrix was not symmetric. The aim of this application was to identify research clusters within a group of 44 academics. The main result was to make it apparent that there was a network of collaborations among the group rather than clear clusters. It was, however, possible to use the results to cluster the individuals more effectively than had been possible from the raw input data of collaboration vectors.

7 Introductions to publications

Here the work on SSNN, and later HASSANN, is expanded on, giving a feel for the published, and to-be-published work that came out of this project.

7.1 Paper 1: "Elucidating Protein Secondary Structure with Circular Dichroism and a Neural Network" by V. Hall, A. Nash, E. Hines, A. Rodger

The first paper is published in the Journal of Computational Chemistry and is on the work of SSNN, Secondary Structure Neural Network. SSNN is introduced as a SOM that clusters CD spectra of globular proteins and their corresponding secondary structures in 6 different structure types: $\alpha-$helix-regular, $\alpha-$helix-distorted, $\beta-$sheet-regular, $\beta-$sheet-distorted, turns, other.

Here SSNN is compared with SELCON3, K2d algorithmic methods for finding protein secondary structures using CD data. There are 48 proteins used to train SSNN here, the same exact spectra and structures that were used to train algorithms on Dichroweb when the reference set 7 is selected.

SSNN is compared with SELCON3 in a LOOCV manner, so the algorithms take on the names SELCON3-47 and SSNN-47 as they are repeatedly trained with 47 spectra and structures. Here K2d is trained with 24 spectra, as we could not find a re-trainable version of K2d, so a K2d-24 comparison was performed. The structure NRMSDs are compared with 3 structure-types, as these are the most that K2d can predict. Summed over all structure types, the NRMSD for SSNN-47 come out best at 10.67, followed by SELCON3-47 with 11.05 then K2d with 12.32. The best helix NRMSDs come from K2d, the best sheet *and* Other NRMSDs from SSNN-47.

A study of which algorithm best predicts proteins that are rich in each structure type is also done. In this test SELCON3-47 is the best predictor for helical

proteins, SSNN-47 for sheet-rich proteins, and also proteins rich in Other structures.

7.2 Paper 2: "Protein Secondary Structure Prediction from Circular Dichroism Spectra Using a self-organising Map with Concentration Correction" by V. Hall, M. Sklepari, A. Rodger

This paper has been accepted by Chirality journal. Here we follow on from the excellent structure prediction capabilities that SSNN achieved earlier in the project, and succeed in making the methodology more robust.

A new tool is built to suggest corrections of reported concentrations for proteins submitted to CD spectrophotometers. The concentration of a protein in solution is of paramount importance for understanding its structure and therefore its function.[43] points out that it is "absolutely essential to have precisely correct concentration measurements (not just estimates from colorimetric assays)..." Greenfield says that least-squares analysis programs are the only options that can be used to estimate concentrations of proteins in solution. This is one reason we developed SSNN to perform concentration correction, now it is possible to get good structure predictions when the concentration is not known accurately.

In this paper, we show how SSNN with concentration-correction has been used to find the correct concentrations of lipoproteins and proteins. The motivation for this is that when dealing with very small weights (micrograms μg) and volumes (micro litres, μL), it is very difficult to accurately measure protein powders and buffer, and other liquid volumes. For this reason, errors easily arise when

making solutions of proteins to work with in the lab. This is done by multiplying the original spectra by scaling factors, effectively scaling the concentrations then submitting them to the test module of SSNN. The CD spectrum is multiplied by numbers usually between 0.1 and 10, this produces spectra that match the spectra SSNN was trained with to greater or smaller amounts. Matching to greater amounts shows that those scaled spectra are closer to the real spectra that are expected by SSNN, and so are more faithful representations of the actual nature of the protein in question. This shows that their concentrations are more accurate than that reported.We find that predictions of proteins with high random coil and extremely high helical content can be improved, so add 5 spectra: 2 theoretical, extrapolated 100 % helical proteins, and 3 truly 100 % random coil proteins. This takes the training set up to 53 spectra.

We study proteins from the laboratory that were previously abandoned due to not knowing their correct structures. We study β-sheet polylysine; a set of 4 lipoproteins; ZapA WT and mutants; some toxins; and Sufl polypeptide in various solvents. Best concentrations are found using NRMSDs plotted against concentrations. With the concentration correction, we manage to obtain fairly reliable structure predictions for all molecules looked at.

We also report in this paper that we made a GUI of SSNN, and uploaded it to the Alison Rodger Group website for free academic use, and licensed commercial use via Warwick Ventures.

7.3 Paper 3: "SSNN, a method for neural network protein secondary structure fitting using circular dichroism data" by V. Hall, A. Nash, A. Rodger

This paper has been published in Analytical Methods journal. Once SSNN became available for download, and use by anyone for their data mining or machine learning research, there needed to be somewhere users of the application could get guidance on installing and running it for examples they could implement for their own protein and CD work.

Here SSNN is compared with CDSSTR, by Curtis Johnson *et al.* as CDSSTR is one of the best CD spectra modelling, and structure prediction methodologies, and we needed to understand how useful SSNN would be in the structure fitting field. Again, we found SSNN compared well, and had additional uses compared with what is available in the field.

The guidelines on how to set-up and run two versions of the SSNN methodology are also included: "SSNNGUI" for quick use by anyone wanting structures from CD spectra, and "SSNN1_2" for re-training the SOM with *any* data set. These include the single run for structure determination, and the multiple run SSNN for concentration correction. This software is given (with examples) for Windows and Mac operating systems. The SSNN applications available online are stand-alone MATLAB GUIs that do not require MATLAB to run, but come with the Compiler Runtime from MATLAB.

Some more example tests of SSNN are done, with reasonable model spectra, and generally good structure predictions resulting.

When comparing with CDSSTR-47, and looking again at previous results for SELCON3-47 and SSNN-47, we find that the additional 5 spectra with 100 % single-structure-type proteins greatly improves the structure estimations given by SSNN-52. SSNN-52 being SSNN trained and tested with 52 spectra in a LOOCV method. SSNN with the expanded training set is now best algorithm overall, as well as for each structure type, although it is joint best with CDSSTR-47 for $\geq$ 30 % $\beta-$sheet proteins.

7.4 Paper 4: "Self organising map pattern recognition for real-time prosthetic control: HASSANN" by V. Hall, M. Ortiz-Catalan

This paper is in preparation. As the capabilities of SSNN grew, we wanted to see how far it could be pushed to achieve more, so it was applied to something very different from CD spectra and protein structures.

This final paper on SSNN's applications, is on the adaptation of the SSNN SOM to myoelectric signal processing for control of robotic arms and hands. This version has the new name "HASSANN", for Hand Activation Signals SOM Artificial Neural Network. The paper briefly describes the BioPatRec software platform that HASSANN is part of.

This paper shows results of offline tests of HASSANN: performing pattern recognition on the MES, and deciding which limb movement is required. An example of the output is shown: a model spectrum is plotted with the myoelectric signals and the residual, also the locations of the BMUs on the SOM.

An RMSE of 0.0371 is obtained for the validation runs, validating the map size, the initial learning rate, the number of BMUs, the neighbourhood size etc. The NRMSE for the model shown is 0.00939, a very good value.

A confusion matrix figure shows how good HASSANN is at predicting movements, and a value telling the same: 0.90 ± 0.08 mean accuracy over the 11 different movements. The range of accuracies is 0.692 to 1.00; this is the value of the number of times HASSANN predicted the movement correctly out of all trials with that movement.

The next work with HASSANN will be for Max Ortiz-Catalan *et al.* to test the capabilities with sequential movements in real-time with people who have incomplete arms, and also people with complete arms. That work will be followed by using HASSANN to recognise diverse movements required simultaneously, so that the limbs can be operated in as natural a motion as possible; sequential movements are not natural, and do not allow much ease of use.

8 Discussion and Conclusions

Every field of science, engineering, business and government produces vast amounts of data these days. Practitioners of circular dichroism spectroscopy have the problem of translating the CD spectra into structures of the molecules they are studying. We set to making a code that could help by learning about CD spectra, and the corresponding structures of proteins, and could become a type of software CD expert that could be copied to anyone who needs more experience in getting structure knowledge from the spectra. Having good knowledge of the structure of a protein is a great help to understanding how it interacts with other proteins, ligands, drugs, pathogens and nucleic acids. This needs to be done for a protein whenever it is used in an experiment, as the molecule will react to every different environment it is placed in. This helps with drug design and quality control.

SSNN, the self-organising map software that was developed for the project was successful in estimating protein secondary structures; protein solution concentrations; and recognising patterns in myoelectric signals, as part of the BioPatRec software platform developed by Max J. Ortiz-Catalan *et al.*, to control robotic arms. For the MES application, the SSNN-derived algorithm took on the name HASSANN.

The SOM architecture is a neural network developed by Teuvo Kohonen in 1982, it has an input layer, and a very large output layer. This output layer is where all the models for the input vectors are clustered, there are no obvious class devisions, rather they have fuzzy boundaries.

8.1 SSNN: CD spectra to structure prediction

SSNN stands for secondary structure neural network, as it takes CD spectroscopy data in the ultraviolet range (240-190 nm), and clusters it to produce estimations of secondary structures of globular proteins and lipoproteins. The fuzzy boundaries from SOM are appropriate, as most proteins and other chiral molecules do not fit into one category, but have varying amounts of each structure type. The exceptions are molecules that are 100 % one structure, or example 100 % random coil.

The dataset that SSNN has been trained with is only 57 by 53 elements, that is light intensity values at 51 wavelengths plus 6 structure types, and 53 CD spectra. Despite the small size of the dataset, this still represents 57 dimensional data, which of course cannot be visualised or understood by a human brain. This is why a neural network approach must be employed.

ML techniques make it easy to cluster the data, and see how spectra are related. Some other teams made neural networks for CD data, so we thought we could go further, and make better spectral models, and structure predictions. Compared with others we used a longer wavelength range than some to get the short-wavelength electronic transition information, we gave our SOM a larger training set, with more coverage of the data space, and more structure types (6) than some. We used a larger map to house this greater range of spectra and conformations. We used more BMUs to construct the models than most research groups in the field of making algorithms for predicting protein secondary structure from CD spectra. We gave a figure of where the BMUs were taken from on the map, and a read-out of which spectra these were. Further, we allowed for concentration errors.

A wavelength range of 240-190 nm was chosen, as it is a widely used range, and due to the electronic transitions being in that region, and shorter wavelengths, but not longer. So there did not appear to be any good arguments for including lower-energy data (wavelengths > 240 nm), as there are only aromatic group transitions in that range, which we are not interested in.

The reason we did not use wavelengths shorter than 190 nm was the difficulty of getting reliable data in that realm without using the rare and expensive synchrotron light sources that are sometimes used for this work (SRCD), where it is possible to get reliable data at least as far as 160 nm.[71] Because most practitioners of CD, in the laboratories where they work regularly, cannot get short wavelengths for their spectra, we concentrated on the 240-190 nm range.

In Papers 1-3 (in subsections 7.1, 7.2, 7.3), and in the Introduction (section 6) we compared SSNN with similar CD-to-structure methodologies: SELCON3, CDSSTR, K2d, SOMCD. Some of these are statistical programs, and others are neural networks.

In Paper 3 we highlighted which methodologies would best be used for which structure types; SSNN-47 was one of the best, and SSNN-52 was better than that.

SSNN may be good, but it is still advisable to use at least a few methodologies for each CD spectrum to see where they agree, and where certain methods are poor or good predictors of particular structures compared with others. This gives a more statistically sound, and reliable estimation of one's protein or lipoprotein structures.

SSNN was also used to correct the concentrations of proteins in solution. This was done by running SSNN repeatedly, scaling the input spectrum with different factors, these factors being what the concentration is relative to the original input spectrum's concentration. So an input spectrum with a reported concentration of 0.1 mg/ml is scaled by 10, and if that turns out to be the correct concentration (smallest NRMSD of all concentrations), then the actual concentration was originally 0.01 mg/ml. The correction is rarely this extreme, it is much more likely to be in the 0.5 to 2.0 range.

Results show that this approach works well, the spectra obtained are very good (Paper 2). The user needs to look carefully at the region with the lowest NRMSD values for the spectrum, it should not be an isolated low point, it should be in the middle of a large depression in NRMSD. If there is a reasonably good model in the 1.0 region, this should be looked at with less scepticism than those far from 1.0, where 1.0 is the original experimental spectrum from the user.

Also, the short wavelength region of the spectrum is more important for determining secondary structures, due to the locations of the electronic transitions. If two spectral models are competing for best, then the spectrum with a more accurate short-wavelength region should be trusted more. Closest attention should be paid to the locations of the electronic transitions that lead to the peaks and troughs in the CD spectrum: does this model predict them well?

Making a SOM give better predictions?

To get a more realistic SOM representation of the data space, one might trial

different map shapes. One approach is to use a SOM that loops around to form a toroidal (doughnut) or spherical shape. We did not attempt this with CD data, but it would probably not have worked well, as the edge structures of proteins in the CD work are extremes, not cyclic, and the edge spectra are not cyclic either. We also did not do this for MES. It is not immediately clear whether the spectra or features for the MES data are cyclic, but the movements are not cyclic.

Where to get the SSNN SOM

SSNN is available on the group website for Windows or Mac, SSNNGUI for pre-trained SSNN with CD spectra from globular proteins, or SSNN1_2 for re-training the SOM with any data set. Instructions on how to use these stand-alone applications are also given, and the software can be licensed by commercial users, or downloaded for free by academics. SSNN is on the A. Rodger Group website here: $http://bit.ly/1p9vbUK$.

8.1.1 SSNN further work

The more reference or training set data that a ML technique has, the better it can be trained to recognise patterns in said data. Of course one must not fall into the trap of over fitting, as touched on in the Introduction, section 6. SSNN-52 made better structure estimates than SSNN-47 did (the final version available on the website has 53 spectra).

The aim must be to increase the reference set for SSNN. Indeed, this could be done if all publicly available CD spectra and protein structures were labelled in the same way. There is no widely-recognised gold standard for secondary structures. A very worthwhile task would be to learn all of the labelling methods,

and then translate the disparate protein structures into one large database. This should enable SSNN and other ML techniques to produce better structure estimations. The secondary structure remarks in the PDB are generated by DSSP. The results are up to interpretation by the person using them: one may decide that residues are slightly different lengths from that suggested by DSSP, so there is no definite ruling.[100]

We would like to train SSNN with transmembrane CD spectra and structures. This would enable its use for such proteins, as they cannot be studied using SSNN trained with globular protein spectra, as it currently is. These structures and spectra are too dissimilar for even reasonable structure estimations or spectral models to be made using the other's reference or training set.

For the application to become adopted by more people we consider it advisable to make a Linux-compatible SSNN available online, perhaps even with a few different Linux OSs, especially as a lot of scientific computing is done in Linux environments these days.

An idea for a great tool would be to gather spectroscopic data from various different spectroscopic sources, such as CD, Raman spectroscopy, ROA (Raman Optical Activity), IR (infrared radiation spectroscopy), and put them all together in a SOM application. There are some techniques that already look at Surface Enhanced Raman Spectroscopy (SERS), IR spectroscopy, and FT-IR data:[101–105]. These techniques use evolutionary algorithms, genetic programming, random forests, and artificial neural networks.

The aim being to have the ability to test a protein with different techniques, to get different views of the same system, comparing the results. This is a similar idea to combining the results from different CD-to-secondary-structure method-

ologies, as discussed above, but with a much wider information base, and with more applications. Of course, using a transmembrane protein database for CD spectra and structures from PDB files (or from some other method like NMR or X-ray crystallography) with the IR, Raman, ROA data source would be appropriate and sensible.

Maybe the application would have one SOM trained with data from all spectroscopies, or likely it would need to possess multiple SOMs one for each structure-determining technique. Perhaps the best approach would be to use various ML techniques, one for each spectroscopic technique, then combine them in one software application.

The concept of having such an intelligent piece of software is exciting; it could be applied to many other fields of research that produce lots of data that need to be interpreted. So many new insights could be gained by looking at essentially the same data from various points of view.

It would be interesting to try other ML techniques for CD spectra training, other than SOM. If there had been more time available, this would be a likely subject to explore. The literature seems to show that more complex ML algorithms are producing more accurate predictions. For example: Jianhua Yang's thesis.[106]

The literature shows that the best way to arrange the algorithms for these more complex forms that produce better results is to use one machine learning technique on the lowest-level of data, and apply its findings to another ML algorithm, repeating this for N levels, each time learning more high-level information. This is called (hierarchical) deep learning.[107]

In his thesis Yang[106] 1) used a GA to evolve the best inputs for a neural network, then 2) used the NN to form models for classification or prediction, then there was a round of mathematical programming that was used to find regression rules. Jianhua Yang is a PhD graduate of the School of Engineering at the University of Warwick. Independent Component Analysis was also used with the NN.

There is some evidence that the human brain learns to arrange information in a hierarchical manner; the n^{th} level of the system extracts a bit of information, then passes the rest of the information, and its own findings on to the next level up.[108] The first level might recognise a short horizontal line, the next might gather information from the outputs from the first level and realise it is looking at a letter "E". The next level sees the letters "England", and realises that is a word, higher levels realise this is a country on an island, part of the continent of Europe. Eventually some level might think that there are a lot of literary works produced from this country in the eponymous language.[109]

Applying this approach to CD spectra work would probably proceed like this: a PCA methodology could be used to feed the principal components to a SOM, which then does the protein conformation estimation more easily.

Or a SOM could be used to find the clustering and structure results, then one could use a genetic algorithm (GA) to evolve SOMs with different parameters like map size and learning rate.

Another approach would be to train various different ML algorithms, of different types with the CD and conformation spectra and knowledge, and combine them in an application to compare the results, then try to understand what char-

acteristics that lead to those conformation findings with higher accuracies. So these ML techniques would all be on the same level of abstraction, they would take in the same data.

However, it is not clear that CD spectra are this difficult to learn: the data are not that expansive, and many structures are known. (Well, perhaps the technique would be useful if there is the additional problem of having protein structures annotated differently with the diverse structure labelling methods). This would better be applied to data mining a vast quantity of unlabelled data that also does not demand low latency.

The aim of a machine learning technique is always to find the best solution, or the solution with the lowest error, or highest correlation coefficient. It stands to reason that a truly clever learning algorithm would expend lots of computing resources to explore an *apparent* global minimum in error or energy usage (much like all other ML techniques), but also keep feelers out in the rest of the data space, giving a small amount of computing time (per square solution space, if you excuse the 2 dimensional analogy) to search in case lower minima are found.

This seems like a strategy that would hedge bets very well, while still finding all of the local minima as the pattern recognition progresses. This way it would be making little successes along the way, but would still end up finding the global best solution. Of course, at the beginning, an acceptable size for the solution space to be explored would have to be decided, so too much computational time is not used.

8.2 HASSANN: myoelectric data for limb movement

HASSANN stands for Hand Activation Signals SOM Artificial Neural Network; it takes pre-processed myoelectric data recorded using electrodes on people's arms. This is very high-dimensional data, so the pre-processing to extract signal features is very important, as it makes the classification much more accurate, and reduces processing time greatly. The pre-processing is done by the earlier stages of BioPatRec. HASSANN is introduced in Paper 4, in subsection 7.4 or reference[9].

The MES (myoelectric signal) data is 36,000 by 4 by 10; 36,000 dimensions is too much to cluster with a SOM. In MATLAB on a laptop it would take weeks to get results, given that the SSNN work took about 20 minutes to run the 57 by 53 element data. The raw MES signal data is 476.66 times the size of the CD data, so running on the same Mac with the same SOM parameters, it would take at least 159 hours, 9 short of a week. That is just for one patient of 20. The program would be comparing very long spectra as well, which would greatly slow many stages of the program training validation, and testing.

There were at least three databases in the BioPatRec repository, previously gathered by the BioPatRec team, the one HASSANN was trained on was 10 movements (besides "rest"), 4 electrode channels, 20 patients, all able-bodied. Other databases use 6 movements and 8 channels, some have simultaneous movements. The work reported on in[98] was approved by the Swedish Regional Ethics Committee in Gothenburg (626-10, T688-12).
It has been suggested that batch training a SOM is faster than using stochastic individual training, and has a higher chance of convergence, but we did not find it necessary with SSNN or HASSANN, the stochastic individual selection

for many thousands of iterations produced maps that made good estimations or predictions. SSNN needed 28,000 iterations, HASSANN only 5,000, but with a slightly higher initial learning rate, L_0: SSNN: $L_0 = 0.06$, HASSANN: $L_0 = 0.1$. The results of this work came without much optimisation, it worked well early on.

The error of SSNN now that it is trained with 53 spectra is very good, better than any methodology it was compared with, and the accuracy of HASSANN's movement recognition is also very high.

The confusion matrix (Figure 13) in the regions not on the diagonal is entirely the first three shades of blue.[9] This shows that movements were recognised incorrectly between 0 and about 0.2 of the time. The worst movement for recognition being movement 3, flex hand with an accuracy of 0.692.

Further work on the methods used to gather the MES data, and how many channels should be used is needed. The team have experimented with 4 or 8 channels. Historically, researchers have found that using too many channels can just lead to confusion with regard to the software, and too much pressure on the computing hardware. However, with this use of pattern recognition software, feature extraction that greatly reduces the difficulty of the problem, coupled with hierarchical deep learning, and ever increasing processing power, the issues of many channels, should they prove helpful, should become less and less daunting.

The feature extraction employed by BioPatRec is not the same as the deep learning mentioned in section 8.1.1 on further work with SSNN. Feature extraction of BioPatRec is more manually coded than that. The programmer decides what features are extracted, rather than some genetic algorithm, PCA or neural network. Some features used are: how many times the signal crosses zero, the

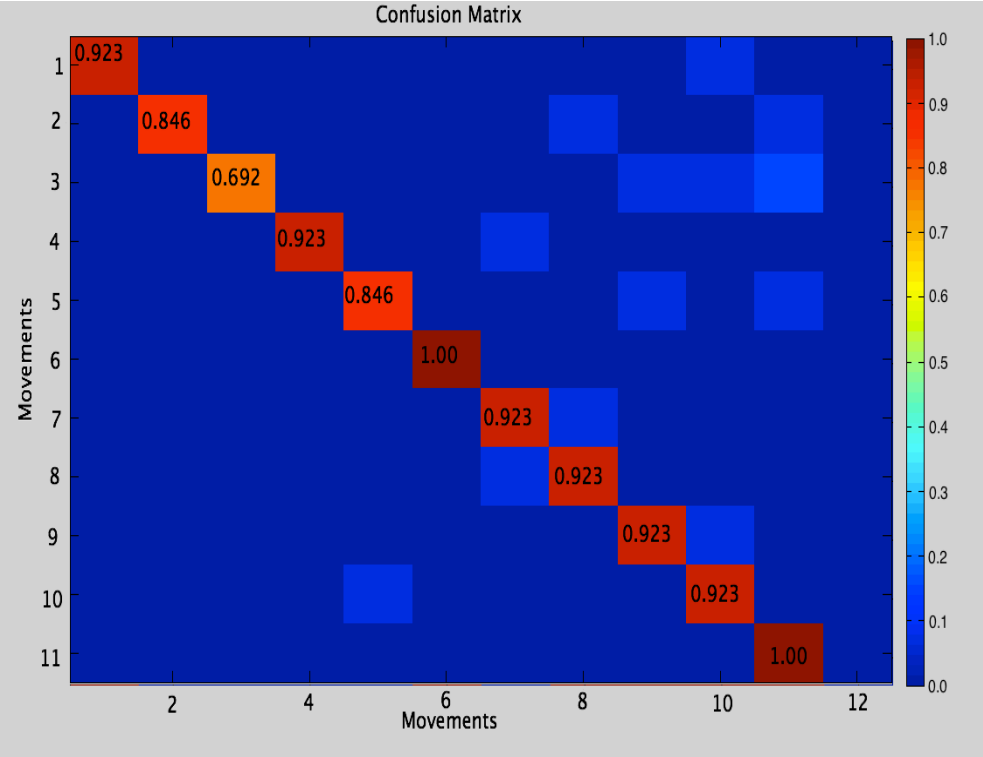


Figure 13: The confusion matrix of HASSANN, reproduced from Paper 4, reference[9].

absolute mean values of the time windows (small divisions), median, variance, standard deviation, the peaks, the mean velocity, the slope changes. These can be found on the BioPatRec website: `https://code.google.com/p/biopatrec/wiki/SigFeatures`.

8.2.1 HASSANN further work

Future work for the HASSANN application would be to test it in real-time, and then to train and test it for simultaneous movements. Perhaps it could be trained with data from all of the patients, once the data set has been divided into training, validation, and test sets, as it was divided for the work in Paper 4.

The aim for BioPatRec is to use it as a platform to compare and improve pattern recognition software for limb control. The hardware is not the focus of the work, as there are various companies and groups working successfully on the hardware side of things. The software is the most difficult component right now, and the industry needs the complexity and accuracy of the software to grow in the coming years to make more capable and easy-to-use prosthetic limbs.

Ultimately, of course, the systems that use the BioPatRec and HASSANN software should perform natural movements with no delay and great reliability. Therefore, any work towards improving prediction accuracy, comfort, speed and or simultaneous actions would be advantageous.

8.3 Final summary

The software reported on in this thesis has been applied to diverse datasets, and has returned very competitive results in the circular dichroism-to-secondary struc-

ture estimation field; providing the correct concentrations of protein solutions; and pattern recognition of myoelectric signals for control of robotic prosthetic limbs.

The SSNN neural network application that is freely available, could be applied to *any* dataset: labelled or unlabelled for supervised learning or unsupervised data mining in data dimensions that cannot be understood by a human brain. As any Machine Learning methodology should, SSNN produces very concise results that condense the pattern recognition findings, while allowing the user to see the big picture of how the data elements are related to each other.

References

[1] W. Kabsch and C. Sander, "Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features," *Biopolymers*, vol. 22, no. 12, pp. 2577–2637, 1983.

[2] "dichroweb.cryst.bbk.ac.uk," 2014.

[3] "http://www.lohninger.com/kohonen.html," 2013.

[4] V. A. Hall, A. Nash, and A. Rodger, "SSNN, a method for neural network protein secondary structure fitting using circular dichroism data," *Analytical Methods*, vol. 6, pp. 6721–6726, June 2014.

[5] B. Nordén, A. Rodger, and T. Dafforn, *Linear Dichroism and Circular Dichroism: A Textbook on Polarized-Light Spectroscopy*. Royal Society of Chemistry, 2010.

[6] "http://www.bmb.ogi.edu/users/jww/mcd02.html."

[7] Magnus Manske, "http://tinyurl.com/l66ey55."

[8] ASSH, "http://www.assh.org/public/handanatomy/pages/default.aspx," 2009.

[9] V. A. Hall and M. Ortiz-Catalan, "Self organising map pattern recognition for real-time prosthetic control: HASSANN." in preparation, 2014.

[10] "http://www.ibm.com/big-data/us/en/."

[11] G. E. Marchant, "The growing gap between emerging technologies and the law," in *The Growing Gap Between Emerging Technologies and Legal-Ethical Oversight* (G. E. Marchant, B. R. Allenby, and J. R. Herkert, eds.),

vol. 7 of *The International Library of Ethics, Law and Technology*, pp. 19–33, Springer Netherlands, 2011.

[12] T. Kohonen, "Self-organised formation of topologically correct feature maps," *Biological cybernetics*, vol. 43, pp. 59–69, January 1982.

[13] P. Unneberg, J. Merelo, P. Chacón, and F. Morán, "Somcd: Method for evaluating protein secondary structure from uv circular dichroism spectra," *Proteins: Structure, Function, and Bioinformatics*, vol. 42, pp. 460–470, January 2001.

[14] T. Kohonen, "Essentials of the self-organizing map," *Neural networks*, vol. 37, pp. 52–65, 2013.

[15] R. Gray, "Vector quantization," *IEEE ASSP Magazine*, vol. 1, pp. 4–29, Apr 1984.

[16] E. Forgy, "Cluster analysis of multivariate data: efficiency vs. interpretability of classification," *Biometrics*, vol. 21, p. 768, 1965.

[17] V. A. Hall, A. Nash, E. Hines, and A. Rodger, "Elucidating protein secondary structure with circular dichroism and a neural network," *Journal of Computational Chemistry*, vol. 34, no. 32, pp. 2774–2786, 2013.

[18] H.-J. Kim and K.-S. Shin, "A hybrid approach based on neural networks and genetic algorithms for detecting temporal patterns in stock markets," *Applied Soft Computing*, vol. 7, no. 2, pp. 569–576, 2007.

[19] D. Howe, M. Costanzo, P. Fey, T. Gojobori, L. Hannick, W. Hide, D. P. Hill, R. Kania, M. Schaeffer, S. St Pierre, *et al.*, "Big data: The future of biocuration," *Nature*, vol. 455, no. 7209, pp. 47–50, 2008.

[20] J. Mena, *Investigative Data Mining for Security and Criminal Detection*. Elsvier Science, 2003.

[21] S. Nelson, "Big data: the harvard computers," *Nature*, vol. 455, no. 7209, pp. 36–37, 2008.

[22] S. Lohr, "The age of big data," *New York Times*, vol. 11, 2012.

[23] W. Wang, L. Chen, K. Thirunarayan, and A. P. Sheth, "Harnessing twitter" big data" for automatic emotion identification," in *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pp. 587–592, IEEE, 2012.

[24] BLOSSOM, "http://www.somj.com the blossom software package," 2005.

[25] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face Recognition: A Convolutional Neural Network Approach," *IEEE Trans. Neur. Net., Special Issue Neur. Net. and Pat. Rec.*, vol. 8, no. 1, pp. 98–113, 1997.

[26] M. A. Andrade, P. Chacón, J. J. Merelo, and F. Morán, "Evaluation of secondary structure of proteins from UV circular dichroism spectra using an unsupervised learning neural network," *Protein Eng.*, vol. 6, pp. 383–390, January 1993.

[27] C. LouisJeune, M. AndradeNavarro, and C. PerezIratxeta, "Prediction of protein secondary structure from circular dichroism using theoretically derived spectra," *Proteins: Structure, Function, and Bioinformatics*, vol. 80, no. 2, pp. 374–381, 2012.

[28] C. Perez-Iratxeta and M. Andrade-Navarro, "K2d2: Estimation of protein

secondary structure from circular dichroism spectra," *BMC Structural Biology*, vol. 8, p. 25, January 2008.

[29] B. Fritzke, "Growing cell structures–a self-organizing network for unsupervised and supervised learning," *Neural networks*, vol. 7, pp. 1441–1460, Mar 1994.

[30] F. Aurenhammer, "Voronoi diagrams–a survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, 1991.

[31] J. Conway and N. Sloane, "Voronoi regions of lattices, second moments of polytopes, and quantization," *Information Theory, IEEE Transactions on*, vol. 28, no. 2, pp. 211–226, 1982.

[32] H. Ritter, "Self-organizing semantic maps," *Biological cybernetics*, vol. 61, pp. 241–254, 1989.

[33] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen, "SOM PAK: The Self-Organizing Map program package http://www.cis.hut.fi/research/som_lvq_pak.shtml," 1996.

[34] J. R. Anderson, R. S. Michalski, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, *Machine learning: An artificial intelligence approach*, vol. 2. Morgan Kaufmann, 1986.

[35] "http://www.thefreedictionary.com," 2014.

[36] L. Kaelbling, "Reinforcement learning: A survey," *J. of Artificial Intelligence Research 4*, pp. 237–285, 1996.

[37] A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 1998.

[38] V. A. Hall, M. Sklepari, and A. Rodger, "Protein secondary structure prediction from circular dichroism spectra using a self–organising map with concentration correction," *Chirality (accepted)*, 2014.

[39] D. H. Wolpert and W. G. Macready, "No free lunch theorems for search," tech. rep., Technical Report SFI-TR-95-02-010, Santa Fe Institute, 1995.

[40] D. Wolpert, "The lack of a priori distinctions between learning algorithms," *Neural Computation*, vol. 8, pp. 1341–1390, 1996.

[41] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, pp. 78–87, Oct 2012.

[42] P. Domingos, "A unified bias-variance decomposition," in *Proceedings of 17th International Conference on Machine Learning. Stanford CA Morgan Kaufmann*, pp. 231–238, 2000.

[43] L. Whitmore, "Protein secondary structure analysis from circular dichroism spectroscopy: methods and reference databases," *Biopolymers*, vol. 89, pp. 392–400, 2007.

[44] G. D. Fasman, "Circular dichroism and the confomational analysis of biomolecules," *Journal of the American Chemical Society*, vol. 118, pp. 12871–12871, December 1996.

[45] P. W. Atkins, *Molecular quantum mechanics*. Oxford University Press, 1983.

[46] P. W. Atkins, *Physical chemistry*. Oxford University Press, 4th ed., 1991.

[47] J. M. Hollas, *Modern Spectroscopy*, vol. 2nd. John Wiley and Sons, 1992.

[48] N. J. Greenfield, "Using circular dichroism collected as a function of temperature to determine the thermodynamics of protein unfolding and binding interactions," *Nature protocols*, vol. 1, pp. 2527 – 2535, Jan 2006.

[49] *Compendium of Chemical Terminology*. IUPAC Nomenclature Books Series (commonly known as the "Colour Books"), IUPAC, 2nd. ed., 1996.

[50] *The Oxford Pocket Dictionary of Current English*. Oxford University Press, 2009.

[51] S. J. Orfanidis, "Electromagnetic Waves and Antennas." unpublished, May 2014, 2008.

[52] R. Scorpio, *Fundamentals of Acids, Bases, Buffers & Their Application to Biochemical Systems*. Kendall/Hunt Publishing Company, 2000.

[53] S. Kelly, T. Jess, and N. Price, "How to study proteins by circular dichroism," *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics*, vol. 1751, no. 2, pp. 119–139, 2005.

[54] L. Stevens, R. Eisenthal, and M. J. Danson, eds., *Buffers and The Determination of Protein Concentration in Enzyme Assays: A Practical Approach*. Oxford University Press, 1992.

[55] N. Sreerama. . . , "Estimation of protein secondary structure from circular dichroism spectra: comparison of contin, selcon, and cdsstr methods with an expanded reference set," *Analytical biochemistry*, Jan 2000.

[56] N. Sreerama, "Estimation of protein secondary structure from circular dichroism spectra: Comparison of contin, selcon, and cdsstr methods with an expanded reference set," *Anal. Biochem.*, vol. 287, pp. 252–260, 2000.

[57] N. J. Greenfield, "Using circular dichroism spectra to estimate protein secondary structure," *Nature protocols 1.6*, pp. 2876–2890, 2007.

[58] N. J. Greenfield, "Methods to estimate the conformation of proteins and polypeptides from circular dichroism data," *Analytical biochemistry*, vol. 235, pp. 1–10, Mar 1996.

[59] W. C. Johnson, "Protein secondary structure and circular dichroism: A practical guide," *Proteins: Structure, Function, and Bioinformatics*, vol. 7, no. 3, pp. 205–214, 1990.

[60] A. Tourmadje, "Extending cd spectra of proteins to 168 nm improves the analysis for secondary structures," *Anal. Biochem.*, vol. 200, no. 2, pp. 321–331, 1992.

[61] G. Böhm, R. Muhr, and R. Jaenicke, "Quantitative analysis of protein far uv circular dichroism spectra by neural networks," *Protein Eng*, vol. 5, pp. 191–5, Apr 1992.

[62] N. Sreerama, "A self-consistent method for the analysis of protein secondary structure from circular dichroism. analyt. biochem.," *Analytical Biochemistry*, vol. 209, pp. 32–44, 1993.

[63] N. Sreerama and R. Woody, "Poly(Pro)II Helixes in Globular Proteins: Identification and Circular Dichroic Analysis," *Biochemistry*, vol. 33, no. 33, pp. 10022–10025, 1994.

[64] N. Sreerama, "Protein secondary structure from circular dichroism spectroscopy," *J. Mol. Biol.*, vol. 242, pp. 497–506, 1994.

[65] C. Chang, C.-S. Wu, and J. Yang, "Circular dichroic analysis of protein

conformation: Inclusion of the beta-turns," *Analytical biochemistry*, vol. 91, no. 1, pp. 13–31, 1978.

[66] J. Yang, C.-S. Wu, H. Martinez, and S. C. H. W. Hirs, "Calculation of protein conformation from circular dichroism," *Methods in Enzymology*, vol. 130, pp. 208–269, 1986.

[67] T. Kohonen, S. Kaski, and H. Lappalainen, "Self-Organized Formation of Various Invariant-Feature Filters in the Adaptive-Subspace SOM," *Neural Computation*, vol. 9, pp. 1321–1344, August 1997.

[68] B. Bulheller and J. Hirst, "Dichrocalc—circular and linear dichroism online," *Bioinformatics*, vol. 25, pp. 539–540, Feb 2009.

[69] RSCB, "http://www.rcsb.org/pdb/home/home.do," 2014.

[70] NCBIStructureGroup, "https://www.ncbi.nlm.nih.gov/structure/index.shtml," 2014.

[71] B. A. Wallace and R. W. Janes, "Synchrotron radiation circular dichroism spectroscopy of proteins: secondary structure, fold recognition and structural genomics," *Current Opinion in Chemical Biology*, vol. 5, no. 5, pp. 567 – 571, 2001.

[72] W. C. Johnson Jr, "Circular dichroism spectroscopy and the vacuum ultraviolet region," *Annual Review of Physical Chemistry*, vol. 29, no. 1, pp. 93–114, 1978.

[73] R. Woody, "Private communication to Bonnie Wallace *et al.*." private communication, unknown year.

[74] A. Abdul-Gader, A. J. Miles, and B. A. Wallace, "A reference dataset for the analyses of membrane protein secondary structures and transmembrane

residues using circular dichroism spectroscopy," *Bioinformatics*, vol. 27, no. 12, pp. 1630–1636, 2011.

[75] "http://dichroweb.cryst.bbk.ac.uk/html/userguide_datasets.shtml," 2014.

[76] K. Englehart, "A robust real-time control scheme for multifunction myoelectric control," *Biomedical Engineering, IEEE Transactions on*, vol. 50, no. 7, pp. 848–854, 2003.

[77] M. Le Blanc, "Give hope – give a hand – the ln-4 prosthetic hand http://www.stanford.edu/class/engr110/2011/leblanc-03a.pdf," 2008.

[78] NHS, "http://www.nhs.uk/conditions/amputation/pages/introduction.aspx," 2012.

[79] "http://www.cdc.gov/ncbddd/birthdefects/data.html," July 2013.

[80] K. Farry, "Myoelectric teleoperation of a complex robotic hand," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 775–788, Oct 1996.

[81] "http://www.chalmers.se/en/pages/default.aspx," 19 May 2014.

[82] "http://www.integrum.se/index.php/research," 2014.

[83] M. Carrozza, "Design of a cybernetic hand for perception and action," *Biological cybernetics*, vol. 95, pp. 629–644, 2006.

[84] D. Gow, "The development of the edinburgh modular arm system," *Proceedings of the Institution of Mechanical Engineers*, vol. 215, pp. 291–298, 2001.

[85] W. Sauter, "Prosthesis with electric elbow and hand for a three-year-old multiply handicapped child," *Prosthet Orthot Int.*, vol. 9, no. 2, pp. 105–108, 1985.

[86] A. Ajiboye, "A heuristic fuzzy logic approach to emg pattern recognition for multifunctional prosthesis control," *IEEE Transactions on Rehabilitation Engineering*, vol. 13, no. 3, pp. 280–291, 2005.

[87] A. Schwartz, "Extraction algorithms for cortical control of arm prosthetics," *Current Opinion in Neurobiology*, vol. 11, pp. 701–707, 2001.

[88] G. Reina, "On the relationship between joing angular velocity and motor cortical discharge during reaching," *J. Neurophysiol*, vol. 85, pp. 2576–2589, 2001.

[89] J. Soechting, "Moving in three-dimensional space: frames of reference, vectors, and coordinate systems.," *Annu Rev. Neurosci.*, vol. 15, pp. 167–191, 1992.

[90] S. H. Tillery, "Task dependence of primate arm prostures," *Exp. Brain Research*, vol. 104, pp. 1–11, 1995.

[91] A. D. o. I. Investopedia, US, "http://www.investopedia.com/terms/a/autoregressive.asp," 2014.

[92] I. WebFinance, "http://www.businessdictionary.com/definition/autoregression.html," 2014.

[93] A. Oppenheim, "From frequenct to quefrency: A history of the cepstrum," *IEEE Signal Processing Magazine*, pp. 95–106, Sep 2004.

[94] M. C. J.-U., "A Real-Time EMG Pattern Recognition System Based on Linear-Nonlinear Feature Projection for a Multifunction Myoelectric Hand," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 11, pp. 2232–2239, 2006.

[95] S.-H. Park, "Emg pattern recognition based on artificial intelligence techniques," *IEEE Transactions on Rehabilitation Engineering*, vol. 6, pp. 400–405, Dec 1998.

[96] B. Hudgins, "A new strategy for multifunction myoelectric control," *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 1, pp. 82–94, 1993.

[97] "http://www2.warwick.ac.uk/fac/sci/chemistry/research/arodger/arodgergroup," April 2014.

[98] M. J. Ortiz-Catalan, R. Branemark, and B. Hakansson, "Biopatrec: A modular research platform for the control of artificial limbs based on pattern recognition algorithms," *Source Code for Biology and Medicine*, vol. 8, no. 11, 2013.

[99] O. Molokwu, "Intelligent systems analysis for plantar pressure data," Master's thesis, School of Engineering, University of Warwick, 2011.

[100] A. Cameron, "Personal communication." unpublished, October 2014.

[101] R. M. Jarvis, W. Rowe, N. R. Yaffe, R. O'Connor, J. D. Knowles, E. W. Blanch, and R. Goodacre, "Multiobjective evolutionary optimisation for surface-enhanced raman scattering," *Analytical and Bioanalytical Chemistry*, vol. 397, no. 5, pp. 1893–1901, 2010.

[102] D. I. Ellis, D. Broadhurst, D. B. Kell, J. J. Rowland, and R. Goodacre, "Rapid and Quantitative Detection of the Microbial Spoilage of Meat by Fourier Transform Infrared Spectroscopy and Machine Learning," *Appl. Environ. Microbiol*, vol. 68, pp. 2822–2828, June 2002.

[103] M. Kinalwa, E. W. Blanch, and A. J. Doig, "Determination of protein fold

class from raman or raman optical activity spectra using random forests," *Protein Science*, vol. 20, no. 10, pp. 1668–1674, 2011.

[104] N. R. Yaffe, A. Almond, and E. W. Blanch, "A new route to carbohydrate secondary and tertiary structure using raman spectroscopy and raman optical activity," *Journal of the American Chemical Society*, vol. 132, no. 31, pp. 10654–10655, 2010.

[105] R. Goodacre, "Explanatory analysis of spectroscopic data using machine learning of simple, interpretable rules," *Vibrational Spectroscopy*, vol. 32, no. 1, pp. 33 – 45, 2003. A collection of Papers Presented at Shedding New Light on Disease: Optical Diagnostics for the New Millennium (SPEC 2002) Reims, France 23-27 June 2002.

[106] J. Yang, *Intelligent Data Mining using Artificial Neural Networks and Genetic Algorithms: Techniques and Applications*. PhD thesis, School of Engineering, University of Warwick, Gibbett Hill Rd. Coventry, CV4 7AL, 2010.

[107] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 1798–1828, Aug. 2013.

[108] P. E. Utgoff and D. J. Stracuzzi, "Many-layered learning," *Neural Computation*, vol. 14, pp. 2497–2529, 2014/05/14 2002.

[109] R. Kurzweil, *The Singularity Is Near: When Humans Transcend Biology*. Penguin Group U.S., 2005.

Elucidating Protein Secondary Structure with Circular Dichroism and a Neural Network

Vincent Hall,[a,b,c] Anthony Nash,[a,d] Evor Hines,[c] and Alison Rodger*[b,e]

Circular dichroism spectroscopy is a quick method for determining the average secondary structures of proteins, probing their interactions with their environment, and aiding drug discovery. This article describes the development of a self-organising map structure-fitting methodology named secondary structure neural network (SSNN) to aid this process and reduce the level of expertise required. SSNN uses a database of spectra from proteins with known X-ray structures; prediction of structures for new proteins is then possible. It has been designed as 3 units: SSNN1 takes spectra for known proteins, clusters them into a map, and SSNN2 creates a matching structure map. SSNN3 places unknown spectra on the map and gives them structure vectors. SSNN3 output illustrates the process and results obtained. We detail the strengths and weaknesses of SSNN and compare it with widely accepted structure fitting programs. Current input format is $\Delta\varepsilon$ per amino acid residue from 240 to 190 nm in 1 nm steps for the known and unknown proteins and a vector summarizing the secondary structure elements of the known proteins. The format is readily modified to include input data with, for example, extended wavelength ranges or different assignment of secondary structures. SSNN can be used either pretrained with a reference set from the CDPro web site (direct application of SSNN3, with the provided output from SSNN1 and SSNN2) or all three modules can be used as required. SSNN3 is available trained (with the reference set of the 48-spectra set used in this work complemented by five additional spectra) at http://www2.warwick.ac.uk/fac/sci/chemistry/research/arodger/arodgergroup/research_intro/instrumentation/ssnn/. © 2013 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.23456

Introduction

The function of proteins and their interactions with other molecules is completely dependent on their structures. One aspect of this is the local secondary structural motifs formed by neighbouring amino acids in a protein. Circular dichroism (CD) spectroscopy is perhaps the simplest technique that is used to estimate the average secondary structure of a protein. CD is useful, for example, in determining how new drugs interact with biomolecules, for example,[1] and how chiral molecules react to different temperatures, for example,[2] and pHs; for example,[3] it can also be used to find reaction rates, for example,[4] and long-term stabilities (shelf lives) of proteins for example,[5] Estimating secondary structures of proteins using CD requires an expert, and a range of programs have been developed to aid this process as discussed later.

The ability of CD to distinguish changes in protein conformation arises from its dependence on how electronic transitions in the polypeptide backbone of a protein, including the peptide bonds, give different absorption spectra for left- and right-circularly polarized UV light. The absorption spectra also vary depending on the protein's conformation. The CD signal, ΔA, is the difference between the absorbances of left- and right-circularly polarized light:

$$\Delta A = A_L - A_R \quad (1)$$

where A_L and A_R are absorbances of the left- and right-circularly polarized light, respectively. To express CD in terms of $\Delta\varepsilon$, the parameter used in the plots in this article, the Beer-Lambert Law is used:

$$\Delta A = \Delta\varepsilon c l \quad (2)$$

where c is the concentration of the sample, and l is the path length.[1] $\Delta\varepsilon$ is the difference between the extinction coefficients for left- and right-circularly polarized light:

$$\Delta\varepsilon = \varepsilon_L - \varepsilon_R \quad (3)$$

[a] V. Hall, A. Nash
Molecular Organisation and Assembly in Cells Doctoral Training Centre, University of Warwick, Coventry, CV4 7AL, United Kingdom
[b] V. Hall, A. Rodger
Department of Chemistry, University of Warwick, Coventry, CV4 7AL, United Kingdom
E-mail: a.rodger@warwick.ac.uk
[c] V. Hall, E. Hines
School of Engineering, University of Warwick, Coventry, CV4 7AL, United Kingdom
[d] A. Nash
Centre for Scientific Computing, University of Warwick, Coventry, CV4 7AL, United Kingdom
[e] A. Rodger
Warwick Centre for Analytical Science, University of Warwick, Coventry, CV4 7AL, United Kingdom
Contract grant sponsor: Engineering and Physical Sciences Research Council (MOAC Doctoral Training Centre); Contract grant number: EP/F500378/1

One of the issues for protein CD spectroscopy is to decide what units to use for $\Delta\varepsilon$. The most common choice is to use units of molar concentration of amino acids (or amide bonds), in other words the molar concentration of protein times the number of amino acids. This then avoids situations such as dimeric proteins having twice the $\Delta\varepsilon$ measured for the equivalent monomer concentration.

Protein secondary structures are fairly well defined due to the rigid nature of the peptide bond and the free rotation about bonds either side of it. A consequence of this is that the CD spectra can be expressed as a weighted sum of the spectra corresponding to different structural motifs. Deconvolution of spectra can, at least in principle, be used to determine the different proportions of secondary structure motifs present in the protein. For example, an α-helix is characterized by a large positive band at 190 nm (part of the $\pi\rightarrow\pi^*$ exciton couplet), and two smaller negative bands at 208 nm (the other $\pi\rightarrow\pi^*$component) and 222 nm ($n\rightarrow\pi^*$). β-sheets give different signals from α-helices and vary from protein to protein presumably dependent on orientation (parallel/antiparallel), the relative size of the sheet, its three-dimensional twist. There are, however, approximate β-sheet signatures: a positive peak between 195 and 202 nm, and a negative signal between 215 and 220 nm. β-turns have a large negative band at 180–190 nm, a positive signal in the 200–205 nm range ($\pi\rightarrow\pi^*$), and a negative signal at 225 nm ($n\rightarrow\pi^*$). The structure often referred to as "random coil" has a negative signal at 200 nm, which is very similar to both the spectrum of a class of β-sheet proteins and poly-proline II spectra.[6] It is now widely accepted that this 200-nm negative band is dominated by contributions from residues with poly (Pro) II-type conformations.[7] CD data are easy to gather, require minimal sample preparation, and the amount of protein required typically required is 10 μg, although it can be as low as 0.3 μg,[8] but the interpretation of the spectra can be challenging, and to quantify the proportions of different structures with any level of accuracy is not possible without carefully designed software.

A number of secondary structure analysis programs exist, for example.[9–12] It is possible to make use of some of these on Dichroweb, an online server hosted at Birkbeck, University of London.[13] Here, researchers may upload CD spectra and receive secondary structure analyses. Most fitting programs are based on patterns of CD spectra from known proteins of known secondary structure. Most of the available programs use either statistical methods or intelligent systems. The commonly used statistical methods which are available on Dichroweb include: CONTIN which is a ridge regression technique; CDSSTR (an update of "VARSLC") which is a variable selection, or feature selection method; and SELCON (now SELCON3) which is a self-consistent method together with a singular value decomposition algorithm. Dichroweb includes one intelligent system approach, called K2d,[12] which is a self organizing map (SOM) neural network approach. SOMs are also called self organizing feature maps or Kohonen maps after the inventor of SOMs, Teuvo Kohonen.[14] Kohonen invented the SOM in 1982, and the K2d code was developed by Andrade et al. in 1993. Although the intelligent systems approach appears to have many advantages, K2d has not gained widespread acceptance in the CD community, perhaps because it originally limited its wavelength range to between 240 and 200 nm and considered only three secondary structure motifs: α-helix, β-sheet, and "random coil" (or "other"). K2d has been revised subsequently to K2D3,[15,16] by generating additional theoretical reference spectra using Dichrocalc.[17] The performance of K2D3 does not seem to provide any significant improvement over K2d. SOMCD,[18] whose authors include members of the original K2d team, adds turns as a structural category, expands the wavelength range down to 190 nm, and enhances the reference set used to train the SOM. It is available only as a pretrained SOM.

The motivation for this work was that the SOM approaches seem to have attractive features, but with what was available in the public domain it was possible neither to test them rigorously nor to develop them further by, for example, adding new members of the reference set. We, therefore, developed a new CD structure fitting SOM: secondary structure neural network (SSNN) based on concepts similar to those of K2d.[12] It is not possible to compare the details of SSNN to K2d and its successors as these details are not available in the literature. In summary, SSNN has three independent pieces of code that operate in sequence.

i. SSNN1: this module takes spectra for a set of proteins of known secondary structure content (the reference set) and trains (organizes) them so that related spectra are put near each other on the map. The map has many more nodes to put spectra in than there are spectra, so the gaps are filled-in with intermediate, virtual spectra. Hence the title self-organizing map.
ii. SSNN2: using vectors of the secondary structure contents of the reference proteins, and the same weighting for the virtual nodes as used for the spectra, protein secondary structures are allocated to all nodes on the SOM, thus constructing the structures map.
iii. SSNN3: a CD spectrum of an unknown protein is input, and the output is an estimate of its secondary structure and a model spectrum.

SSNN1 and SSNN2 need only be performed once for a given reference set.

The aim of this work was to develop a SOM to predict protein secondary structure from CD spectra using any chosen wavelength range and any chosen basis set of spectra with associated secondary structure compositions. Various elements of the SOM were modified in an attempt to improve its performance: map size, number of training iterations, the learning rule, neighbourhood size, wavelength range, and the number of best matching units (BMUs). The work includes an extensive analysis of how SSNN compares to existing CD-to-secondary-structure programs including the widely used statistical program SELCON3 and also K2d to determine if the SOM approach has any advantages over statistical approaches.

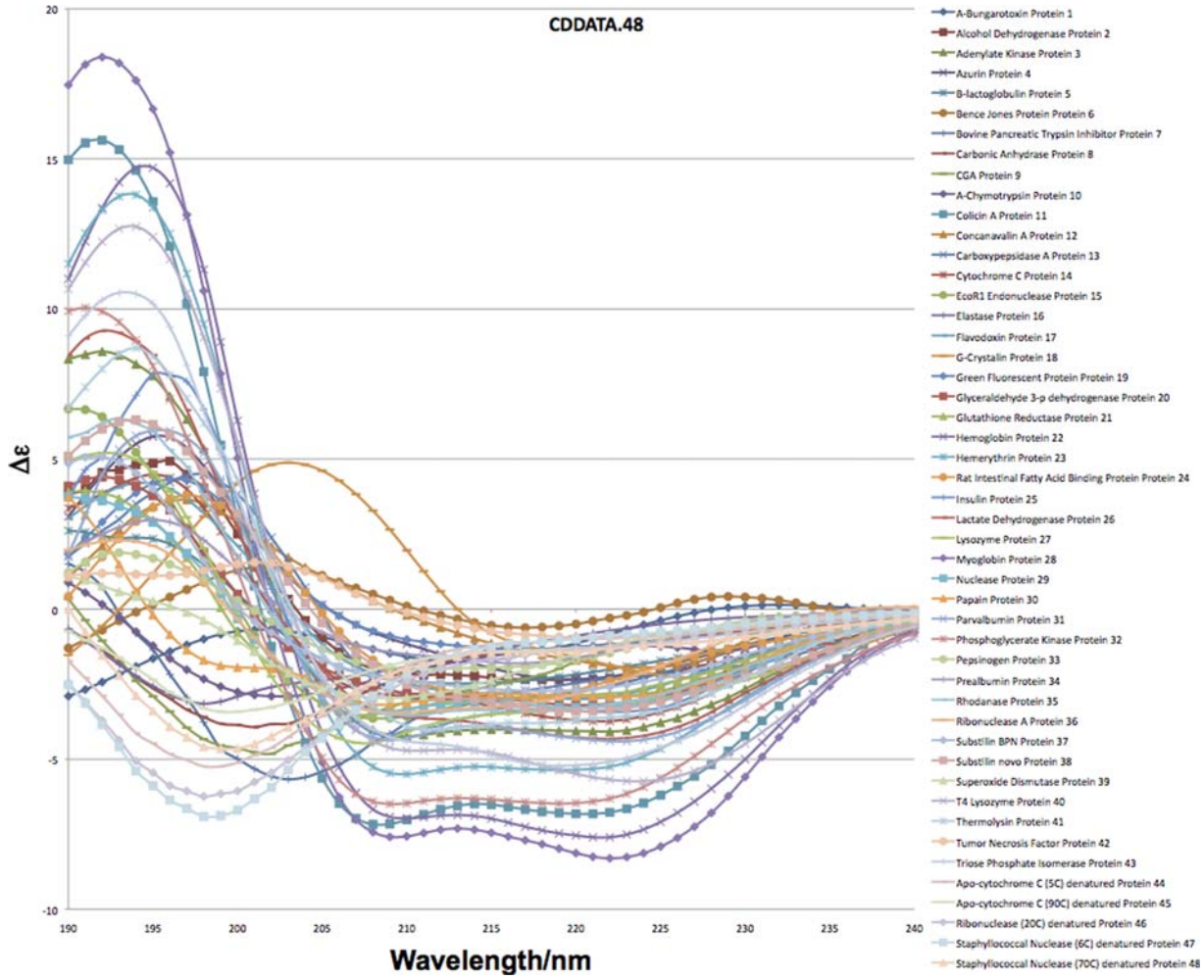


Figure 1. Plot of spectra in the CDDATA.48 reference set.[19]

Methods

Reference data set

The reference data set used to train SSNN was taken from the CDPro website: http://lamar.colostate.edu/∼sreeram/CDPro/. This reference set has been developed by various researchers, and compiled by CDPro and denoted CDDATA.48. The data are the same data as is used on Dichroweb: http://dichroweb.cryst.bbk.ac.uk/html/home.shtml as Reference set 7.[6] The spectra are given in per residue molar absorbance units ($\Delta\varepsilon = \text{mol}^{-1}\ \text{dm}^3\ \text{cm}^{-1}$). This is the largest available reference set that has been consistently annotated with secondary structures. Figure 1 shows an overlay plot of the members of CDDATA.48 showing how it covers the spectra space. The input reference set has 48 proteins with 57 numbers each. In order they are: 51 for the spectrum, and six for the structure values in the order: (α-helix regular, α-helix distorted, β-sheet regular, β-sheet distorted, turns, and other structures). The CDPro website and its references explain the structure types which are summarized.[19] In the reference set there is nonuniform representation of the structure space, for example, the data set contains more α-helix-rich proteins than other structures. The view might be that this would lead to overfitting the SOM for α-helix-rich proteins. However, this is not the case as the proteins in the data set are unrelated. Some of them have similar secondary structure percentages but different sequences, folds, and arrangement of the secondary structure. None of the spectra are identical. Some will, for example, have 35% α-helix, but that could be in one large helix, or in three helices that make up 35% of the protein, or a variety of other arrangements. The CD spectra of these are therefore different.

Although the proteins in CDDATA.48 are soluble globular proteins and we only use the data from 240 to 190 nm, SSNN is not limited in this way. SOMs and neural networks in general are applied to many diverse fields and databases, so we expect SSNN would cope with different reference sets for different classes of proteins. The most important feature of any reference set used with SSNN (or any fitting program) is that all the spectra in a reference set must have their secondary structure vectors determined by the same methodology. In this work, we have shown that CD data over the wavelength range 190–240 nm can give reliable results if a large reference set is used,[19] so we do not anticipate problems from this wavelength range.

In the leave-one-out tests reported in this article, each spectrum in the reference set was removed in turn and used as

the test spectrum. SELCON3 or SSNN1 and SSNN2 was/were run with the resulting 47-member reference set.

Outline of a SOM and SSNN1

A SOM is usually an array, a square, or hexagonal lattice of "nodes," that in some way represents the input data. SSNN1 creates a square SOM from the input CD reference set. Map sizes are named by the length of one side of the square grid, for example, a map size of 40 has 40 × 40 nodes = 1600 nodes. Each node holds a vector of numbers, called a weight vector. The initial SOM has vectors of N random numbers between zero and one at each node, where N is the number of data points in a CD spectrum (in this work 1 datum per nm, so data from 240 to 190 nm is a 51 component vector). The way the SOM is trained is to adjust the weight vector at each node to minimize its distance from each input vector considered in turn. This process changes the weight vectors in a way similar to clustering. In the end, the weight vectors come to mimic the reference set vectors, with the input vectors with vectors that are similar close to each other, and dissimilar ones far apart. Specifically, at each iteration of the training process, one of the input reference set protein spectra is compared to each of the weight vectors to find the most similar weight vector for that spectrum; this is done by calculating a Euclidean distance between each input vector and each nodal vector. The weight vector or "node" with the smallest Euclidean distance from a given input vector from the reference set is given the name "best matching unit," or BMU*. Next a neighborhood of that BMU* is defined. Initially the neighborhood nodes will be random vectors with similar x and y coordinates in the weight map (spectra map). After the weight map has been trained, the neighborhood nodes will contain spectra very similar to that of the BMU* at the center of the neighborhood. At each step, each weight vector of this neighborhood is brought a little closer to the BMU* by a factor of L, a distance-dependent learning rate, in our case

$$\text{Learning}\,(t) = L_0 e^{(-k_1 t)} \tag{4}$$

where L_0 is the initial learning rate, t is the iteration number, and k_1 is a measure of how fast the learning rate decreases. The neighborhood radius changes at each iteration using an equation; SSNN's is

$$r(t) = \begin{cases} (r_0 - 1)\left(1 - \dfrac{t}{t_1}\right) & \text{if } t \le t_1 \\ 1 & \text{if } t > t_1 \end{cases} \tag{5}$$

where r is the radius, r_0 is the initial radius, usually half the size of the map, t_1 is the point when the radius of the neighborhood reaches one, it then stays at one until the end of training. The further away a neighboring weight vector is, the less influence and BMU* will have. This is repeated for a number of iterations, usually thousands.

Characteristics of SSNN1

The SSNN SOM is thus a square lattice of nodes each containing 51 points that are the CD spectra—one point for each wavelength in the range 240–190 nm. SSNN1 was run for 28,000 training iterations. For SSNN, t_1 was set to 7000 iterations, k_1 was 5×10^{-6}, r_0, the initial radius, was set to 20, for a map size of 40 (length of one side). The vectors that are not the BMU*s of real protein spectra, are virtual protein spectra derived as interpolations between the real protein spectra.

The structures map: SSNN2

SSNN1 completes the construction (training) of the SOM, but SSNN includes another step: the assigning of structures step, which involves the creation of a second map, the structures map, and forms the module SSNN2. This is done by constructing another map of 40 × 40 nodes. Each node contains a vector of the fractions of each secondary structure corresponding to its spectral vector. Our reference set of spectra has six secondary structure elements assigned to each protein, but the structure vectors in the structures map have seven numbers. The first six numbers are assigned values corresponding to the fraction of each structure type in the protein (e.g., if 40 amino acids are assigned as α-helices (of the "regular" category) in a 100 amino acid protein, the first entry of that vector will be 0.40). The seventh number is to say whether it is a BMU* of (i.e., essentially the same as) a real protein spectrum; the number of these will match the reference set size. To add structure vectors of the reference set proteins to the SSNN2 map, we start by locating the BMU* for each of the protein spectra in the input reference set on the SSNN1 spectra map. The corresponding nodes in the SSNN2 map are given the structure values for each reference set protein. The other nodes in the map are given structure values by calculating the Euclidean distances between the spectrum of the node and the BMU*s in the SSNN1 map then giving them a structure vector as a weighted sum of a number of nearest BMU* structure vectors with the weighting being the inverse of the distances

$$S_i = \sum_{n=1}^{5} B_n^i \left(\frac{1}{d_n}\right) \tag{6}$$

where $1 \le i \le 6$, S_i is the structure vector component for structure type i, B_n^i is the ith component of the structure vector for the nth BMU*, d_n is the Euclidean distance from the input spectrum to the nth BMU*. For the 48-spectrum reference set, n is optimized at five as discussed in the results and discussion section.

Making the predictions: SSNN3

Once the SOM has a spectra map (from SSNN1) and a structures map (from SSNN2), it can be used to assign secondary structure estimates of proteins with known CD spectra, but unknown structure. SSNN3 locates where the protein with unknown structure would lie on the SSNN1 map, and gives it a structure assignment based on a weighted sum of the structure vectors of the five closest nodes (its BMUs) in the same way as structures are assigned to nodes in SSNN2.

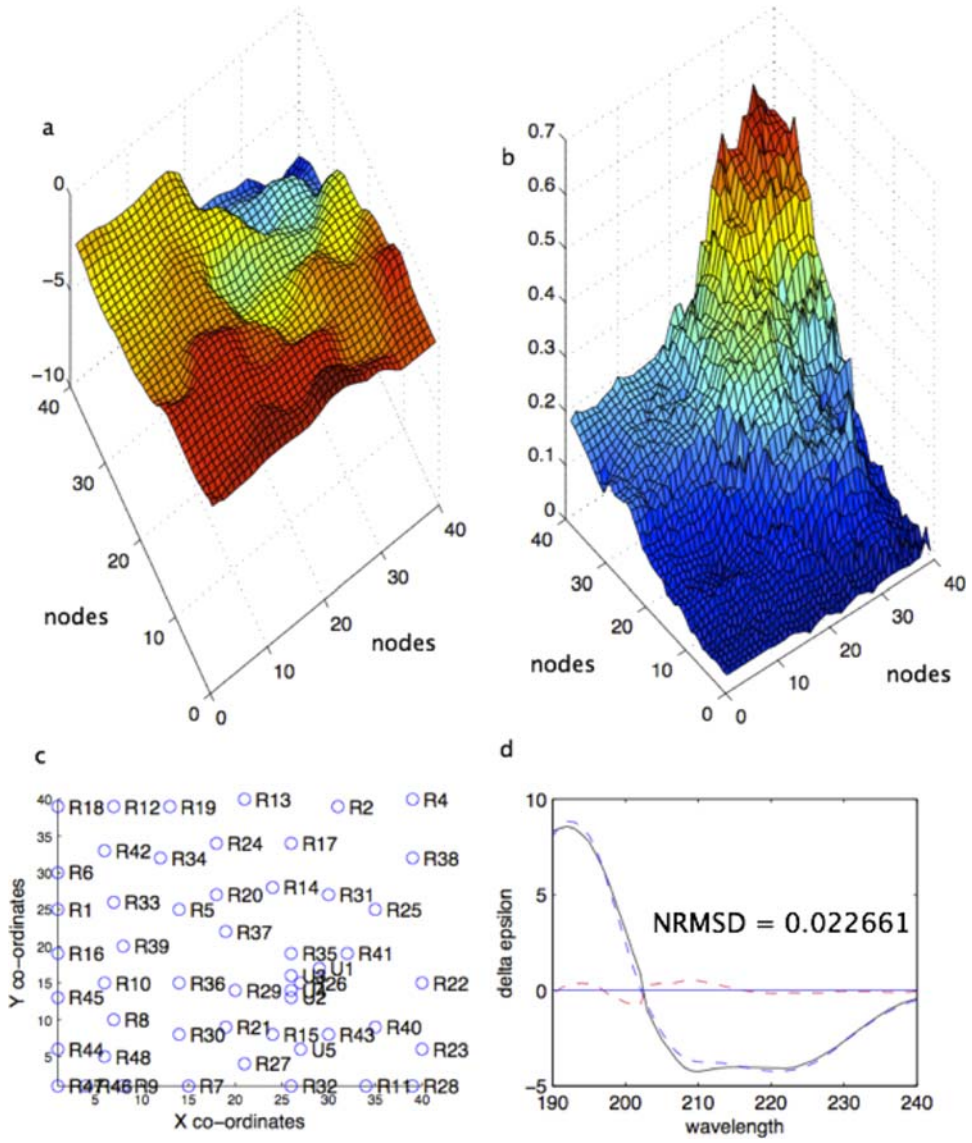


Figure 2. Example of SSNN output, in this case for alcohol dehydrogenase (protein 2 in the reference set, using a reference set with alcohol dehydrogenase excluded): (a) SSNN1: the spectra map at 222 nm (red is high intensity, blue is low); (b) SSNN2: the structures map for α-helix regular (red is high intensity, blue is low); (c) SSNN3: the locations of the BMUs of the unknown protein; (d) SSNN3: the model spectrum (blue dashed line) overlaid on the unknown input spectrum (solid black), with the residual (red dashed line) and the spectral NRMSD. Parameters: number of iterations = 28,000, map size = 40 × 40, initial neighborhood size = 20, $L_0 = 0.1$, $t_1 = 7{,}000$, $k_1 = 5 \times 10^{-6}$, SSNN2 BMUs = 5, SSNN3 BMUs = 5.

An example of the output from SSNN is illustrated in Figure 2. It includes

- **a.** SSNN1: a 3D surface plot of the spectra map at a chosen wavelength (currently set to 222 nm)
- **b.** SSNN2: a similar plot of the structures map for one structure type (currently set to α-helix regular)
- **c.** SSNN3: the positions of the 5 BMUs on the maps in (a) and (b), along with the positions of the reference set proteins
- **d.** SSNN3: an overlay of the unknown and model spectrum together with the difference between them and the spectral normalized root mean square deviation (NRMSD) defined in this work (but not in Dichroweb)[6] as
- **e.**

$$\mathrm{NRMSD} = \frac{\sqrt{\left(\frac{\sum_i \left(x_{i,\mathrm{experiment}} - x_{i,\mathrm{model}}\right)^2}{N}\right)}}{M-m} \tag{7}$$

where x_i is the value at each wavelength, N is the number of data points, M is the largest intensity, and m is the smallest, so $M–m$ is the range. (The structural NMRSD is defined in the same way—but can only be calculated when we know what the "answer" should be.) In addition to this output, there is a text file with the structure vector.

Testing SSNN

The first stage of testing SSNN involved training SSNN1 with 47 of the 48 reference set proteins, and testing its performance on that missing protein. This was repeated for each of the 48 proteins in the reference set (thus retraining the SOM each time), and is called the leave-one-out method or leave-one-out cross-validation (LOOCV). LOOCV is a kind of *k*-fold cross-validation (this *k* refers to the size of reference set), and has been found to be nearly unbiased in its estimate of the true generalization ability of the model being evaluated.[20] All tests were run with a wavelength range of 240–190 nm, except for the wavelength range test. In the final version, SSNN1 was trained for 28,000 iterations, which we found ensured convergence. Convergence was measured by the distortion, which is a sum of all the distances from each reference set spectrum to its BMU, becoming constant. This is a standard test for the performance of a SOM (data not shown).

Results and Discussion

SSNN modules

SSNN1 the SOM-training module, SSNN2, the structure assignment module, and SSNN3, were written in MATLAB. SSNN3 is available trained (with a reference set of the 48-spectra set used in this work complemented by five additional spectra) at http://www2.warwick.ac.uk/fac/sci/chemistry/research/arodger/arodgergroup/research_intro/instrumentation/ssnn/. An example of the pictorial output is given in Figures 2c and 2d. This is accompanied by a text file with the predicted structures in order: (α-helix regular, α-helix distorted, β-sheet regular, β-sheet distorted, turns, and other structures). Before assessing the fitting efficacy of SSNN in general, it was necessary to decide the parameter values to be used. The map size was first varied to see if the SOM needed more data-space to explore for interpolations between protein spectra, or if it would help to reduce the complication by reducing the map size. The number of BMUs was then varied. The learning rate equation was changed from linear to exponential, and the initial learning rate was varied. The wavelength range was then reduced to check the conclusion from other methods that a smaller range reduced the spectra information content too much. Greater and greater errors in concentration were then introduced to see how robust SSNN is against this common experimental error. Errors in wavelength were also introduced to simulate the problems due to incorrectly calibrated CD machines. Finally, comparison of fitting performance to SELCON3, K2d, and SOMCD was performed.

Map size

The average spectral NRMSDs as a function of map size showed a nonlinear decrease (Supporting Information, Fig. S1) in error. Although, "bigger is better" for this parameter, bigger also means more computation time, so we chose a map size of 40 as optimized for most of the subsequent calculations (unless we needed to save computer time) because after this point (larger than 40) the improvement was marginal.

Wavelength range

It is widely accepted that using data only down to 200 nm (as done by K2d) does not give enough spectral information for a good structural estimation unless the protein is either highly helical or has no helical structure. We tested SSNN to see what effect a reduced wavelength range had on the quality of the fit. Supporting Information, Table S1 shows the structural NRMSDs for three different wavelength ranges. The "error bars" are a standard deviation of the NRMSDs, so measure the variation rather than an error. As expected the structural NRMSDs improve significantly with more data, being: 0.14 for data to 200 nm and 0.10 with data to 190 nm. The variations in the NRMSDs increase when shorter wavelengths are included, probably due to the decreasing quality of the data at higher energies.[9] For this work, we selected our wavelength range to be 240–190 nm (although SSNN in not restricted in this way).

Number of BMUs

Comparison of structural NRMSDs as a function of number of BMUs in SSNN3 (Supporting Information, Fig. S2) showed the lowest NRMSDs for 5, with a slight improvement in average fit observed for increasing the number of BMUs from 2 to 5, and little change from 5 to 25. Performance was worse above 25 BMUs. We were initially surprised at this having expected again that more would be better. However, consideration of area to distance ratios leads to the realization that at a larger radius from the first BMU there are many more nodes than at a smaller radius. This could cause distant nodes to have a large effect on the construction of the model thus reducing the quality of predictions. For K2d with a reference set of size 24, Andrade et al. found that 2 was the best number of BMUs to use,[12] whereas for K2D3 with a much larger map, 400 BMUs was adopted.[16] We, therefore, conclude that the optimal number of BMUs is dependent on the size of the reference set and map and should be chosen to ensure the BMUs do cluster about unknown spectra (test spectra) on the map.

Learning rate

An initial learning rate of $L_0 = 0.06$ produced the lowest structural NRMSD of all L_0s tested in the range 0.01–0.8 (Supporting Information, Fig. S4). $L_0 = 0.1$ is only slightly worse than 0.06 and converged more quickly so was generally adopted. Noticeably larger L_0s likely move the nodes in the map to local best approximations of the spectra too early in the training, rather than allowing the map to fully explore the data-space for better approximations. The learning rate was allowed to exponentially decrease with iterations in accord with eq. (4).

Concentration errors

In practice, the biggest problem with protein sample preparation for CD spectroscopy is determining the concentration

accurately. In our experience of various statistical methodologies, errors in concentration for helical proteins translate into similar errors in α-helical estimates. Concentrations determined, for example, by BioRad assays or $A_{280\ nm}$ measurements,[21] give declared protein concentrations that are usually within $\pm$20%. As shown in Supporting Information Figure S5, concentrations errors do worsen the SSNN fit with underestimates being worse than overestimates. As the error is less for an overestimate of protein concentration, erring in this direction is to be advised. It should be noted that K2d is reputed not to need the concentration of the protein solution as input. However, both K2d and SSNN (as currently configured) are based on assuming all reference set and unknown spectra are in $\Delta\varepsilon$ values for molar concentration of amino acids. They, therefore, do need accurate concentrations.

Wavelength errors

Specifications for CD instruments usually accept wavelength errors of up to 1 nm and operators may well be even more relaxed about this parameter. We, therefore, considered the effect of wavelength error on the quality of structural estimates, as summarized in (Supporting Information, Fig. S6). In summary, "blue" (to shorter wavelength) shifts are worse than "red" shifts (toward the visible) with the average error for a 5-nm blue-shift giving approximately the same NRMSD as a 10-nm red-shift. The 2-nm shifts seem to produce no more error than when using a correctly calibrated machine. A 10-nm blue shift is worse than a 10-nm red shift, most likely because a blue shift loses important electronic transition information represented by the intensity of the CD signal in the short wavelength region, while the long wavelength region is relatively flat, with no peaks and little information content.

Comparison of SSNN with SELCON3

Three structure fitting programs are widely used: CONTIN, CDSSTR, and SELCON3, which are all implemented on Dichroweb.[6] Sreerama and Woody have performed a comprehensive comparison of the three codes, so in this work we compared SELCON3 to SSNN. We used the executable version of its code available on the CDPro website (http://lamar.colostate.edu/~sreeram/CDPro/main.html) so that we could run it in a LOOCV manner which we denote SELCON3-47, to compare directly with SSNN-47. The estimated structure vectors and a comparison with the real values are given in Table 1 for the two methods. The SSNN plots for the 48 proteins are given in the Supporting Information, Figures S7–S54.

The overall performance of SSNN-47 and SELCON3-47 (see average values in the final rows of the table) is similar: SELCON3-47 "wins" for 23 out of 48 spectra and SSNN-47 "wins" for 25. On average, when working with six structure types, one might prefer SELCON3-47 for α-helix and other structure estimates and SSNN-47 for β-sheets and turns. However, if one considers high and low percentages of a particular secondary structure type their average performance ranking changes (Table 2).

For both SELCON3-47 and SSNN-47, of the structure types considered here, the most difficult types to predict are mixed α/β proteins (30–50% α-helix) and high β-sheet (>30%). SELCON3-47 gives structural NRMSDs of 0.3 ± 0.3 and 0.3 ± 0.2, respectively, and SSNN-47 gives 0.2 ± 0.2 for both (errors indicate variations in fit as noted above). Inspection of the overlay of model spectra with experimental data proteins suggest this is in part due to the overlap of the α-helix and β-sheet π-π^* transitions around 190–195 nm and in part due to the comparatively small contribution to the spectral NRMSDs of any error in the 215-nm region of the spectrum, but the significant contribution to structural error indicated by this region. The difference between a single β-sheet negative maximum at 215–219 nm and two α-helix negative maxima at 208 and 222 nm is obvious to the human eye but does not make much contribution in the simple error metric (NRMSD) used in both SELCON and SSNN. So we still recommend complementing fitting programs with a visual inspection of the overlay of experimental data and model spectrum.

Proteins with lots of turns are also hard to analyze, as such spectra are extremely varied and give high spectral NRMSDs, although the average error in prediction is not high. The structure type called "other" is really an amalgam of all the structure types that do not fit into any of the previous structure-type categories. Due to this, one might not expect this category to be predicted with accuracy, and yet the mean NRMSD is reasonable indicating that a sufficiently varied spectral/structural landscape is included in the reference set.

Overall, the best way to classify the structure of a new protein is to use a few different methodologies to find whether they confirm each other's predictions. If they disagree, then it is likely that better data need to be collected—whether that be concentration determination or spectra quality.

Comparison of SSNN with K2d

We compared K2d to SSNN-47 and SELCON-47 using the pretrained version available on Dichroweb that uses a smaller reference set and a wavelength range limited to 200 nm.[12] K2d uses a 13×13 grid of nodes, 2 BMUs, 41 nm of spectral data (240–200 nm), and three structure categories. Unfortunately, we did not have access to the database used for K2d to run SSNN and K2d back-to-back, so the K2d reference set included some of the test proteins, which means its apparent performance is artificially enhanced. As K2d uses only three structure types, SELCON3-47 and SSNN-47 structural predictions were put into three-structure format by summing the two α-helix predictions, summing the two β-sheet types, and including turns into the other category. The results are summarized in Table 3 and given in more detail in Supporting Information, Table S2. With this gathering of structure types, overall SSNN-47 performs significantly better than the other two and K2d is worse than SELCON3-47. K2d (with the advantage of test proteins in the training set) performs best for the α-helical structures. SSNN-47 is best for β-turns and other.

Comparison of SSNN to SOMCD

Our attempts to compare properly to SOMCD[18] were defeated by its lack of availability in any form except a final

Table 1. Structure vectors outputted from SSNN-47 and SELCON3–47 for all proteins in CDDATA.48 from the CDPro web site.

Protein number	Protein name	Method	α-regular	α-distorted	β-regular	β-distorted	Turns	Other	Structural NRMSD	Fit minus real: α-regular	Fit minus real: α-distorted	Fit minus real: β-regular	Fit minus real: β-distorted	Fit minus real: turns	Fit minus real: other	SELCON3 NRMSD - SSNN NRMSD
1	α-Bungarotoxin	Real	0.000	0.000	0.014	0.095	0.284	0.608								
		SELCON3-47	0.003	0.032	0.299	0.129	0.155	0.388	0.409	0.003	0.032	0.285	0.034	−0.129	−0.220	0.112
		SSNN-47	0.018	0.056	0.211	0.121	0.166	0.428	0.297	0.018	0.056	0.197	0.026	−0.118	−0.180	
2	Alcohol Dehydrogenase	Real	0.139	0.115	0.139	0.096	0.214	0.297								
		SELCON3-47	0.131	0.110	0.131	0.078	0.231	0.297	0.052	−0.008	−0.005	−0.008	−0.018	0.017	0.000	−0.142
		SSNN-47	0.149	0.111	0.181	0.093	0.224	0.242	0.193	0.010	−0.004	0.042	−0.003	0.010	−0.055	
3	Adenylate Kinase	Real	0.340	0.206	0.077	0.052	0.012	0.313								
		SELCON3-47	0.261	0.158	0.078	0.064	0.178	0.258	0.411	−0.079	−0.048	0.001	0.012	0.166	−0.055	−0.032
		SSNN-47	0.243	0.162	0.073	0.072	0.185	0.265	0.443	−0.097	−0.044	−0.004	0.020	0.173	−0.048	
4	Azurin	Real	0.047	0.062	0.141	0.109	0.312	0.328								
		SELCON3-47	0.130	0.104	0.169	0.090	0.230	0.288	0.278	0.083	0.042	0.028	−0.019	−0.082	−0.040	−0.041
		SSNN-47	0.142	0.110	0.142	0.097	0.229	0.280	0.319	0.095	0.048	0.001	−0.012	−0.083	−0.048	
5	β-lactoglobulin	Real	0.056	0.111	0.287	0.123	0.216	0.207								
		SELCON3-47	0.101	0.081	0.171	0.100	0.212	0.321	0.294	0.045	−0.030	−0.116	−0.023	−0.004	0.114	−0.014
		SSNN-47	0.111	0.097	0.186	0.120	0.212	0.274	0.308	0.055	−0.014	−0.101	−0.003	−0.004	0.067	
6	Bence Jones Protein	Real	0.000	0.028	0.294	0.196	0.229	0.252								
		SELCON3-47	−0.006	0.009	0.164	0.117	0.200	0.539	0.245	−0.006	−0.019	−0.130	−0.079	−0.029	0.287	0.129
		SSNN-47	0.023	0.035	0.311	0.139	0.199	0.293	0.116	0.023	0.007	0.017	−0.057	−0.030	0.041	
7	Bovine Pancreatic Trypsin Inhibitor	Real	0.069	0.138	0.172	0.069	0.190	0.362								
		SELCON3-47	0.091	0.101	0.152	0.100	0.235	0.304	0.178	0.022	−0.037	−0.020	0.031	0.045	−0.058	0.077
		SSNN-47	0.052	0.071	0.180	0.100	0.208	0.389	0.100	−0.017	−0.067	0.008	0.031	0.018	0.027	
8	Carbonic Anhydrase	Real	0.058	0.104	0.170	0.116	0.240	0.312								
		SELCON3-47	0.031	0.044	0.113	0.061	0.109	0.619	0.243	−0.027	−0.060	−0.057	−0.055	−0.131	0.307	−0.043
		SSNN-47	0.028	0.032	0.059	0.039	0.059	0.784	0.286	−0.030	−0.072	−0.111	−0.077	−0.181	0.472	
9	CGA	Real	0.053	0.082	0.210	0.110	0.210	0.335								
		SELCON3-47	0.049	0.072	0.096	0.060	0.138	0.586	0.220	−0.004	−0.010	−0.114	−0.050	−0.072	0.251	−0.029
		SSNN-47	0.050	0.065	0.077	0.049	0.099	0.660	0.249	−0.003	−0.017	−0.133	−0.061	−0.111	0.325	
10	α-Chymotrypsin	Real	0.069	0.045	0.208	0.106	0.200	0.371								
		SELCON3-47	0.028	0.041	0.153	0.073	0.142	0.544	0.156	−0.041	−0.004	−0.055	−0.033	−0.058	0.173	0.049
		SSNN-47	0.033	0.074	0.176	0.097	0.161	0.459	0.107	−0.036	0.029	−0.032	−0.009	−0.039	0.088	
11	Colicin A	Real	0.529	0.225	0.000	0.000	0.044	0.202								
		SELCON3-47	0.479	0.215	−0.005	0.012	0.113	0.205	0.073	−0.050	−0.010	−0.005	0.012	0.069	0.003	0.008
		SSNN-47	0.486	0.214	0.016	0.013	0.095	0.176	0.065	−0.043	−0.011	0.016	0.013	0.051	−0.026	
12	Concanavalin A	Real	0.000	0.038	0.329	0.135	0.236	0.262								
		SELCON3-47	0.029	0.059	0.246	0.128	0.209	0.315	0.155	0.029	0.021	−0.083	−0.007	−0.027	0.053	0.062
		SSNN-47	0.012	0.053	0.332	0.124	0.178	0.301	0.093	0.012	0.015	0.003	−0.011	−0.058	0.039	
13	Carboxypepsidase A	Real	0.254	0.127	0.111	0.052	0.212	0.244								
		SELCON3-47	0.120	0.113	0.251	0.110	0.173	0.229	0.599	−0.134	−0.014	0.140	0.058	−0.039	−0.015	0.065
		SSNN-47	0.088	0.080	0.263	0.117	0.212	0.239	0.535	−0.166	−0.047	0.152	0.065	0.000	−0.005	
14	Cytochrome C	Real	0.214	0.194	0.000	0.000	0.233	0.359								
		SELCON3-47	0.182	0.152	0.093	0.081	0.228	0.273	0.339	−0.032	−0.042	0.093	0.081	−0.005	−0.086	−0.008
		SSNN-47	0.173	0.122	0.101	0.089	0.224	0.292	0.347	−0.041	−0.072	0.101	0.089	−0.009	−0.067	
15	EcoR1 Endonuclease	Real	0.192	0.127	0.098	0.080	0.210	0.293								
		SELCON3-47	0.180	0.147	0.091	0.079	0.202	0.297	0.049	−0.012	0.020	−0.007	−0.001	−0.008	0.004	−0.124

Table 1. (*Continued*)

Protein number	Protein name	Method	α-regular	α-distorted	β-regular	β-distorted	Turns	Other	Structural NRMSD	Fit minus real: α-regular	Fit minus real: α-distorted	Fit minus real: β-regular	Fit minus real: β-distorted	Fit minus real: turns	Fit minus real: other	SELCON3 NRMSD - SSNN NRMSD
		SSNN-47	0.189	0.181	0.052	0.060	0.253	0.265	0.173	−0.003	0.054	−0.046	−0.020	0.043	−0.028	
16	Elastase	Real	0.021	0.087	0.225	0.117	0.208	0.342								
		SELCON3-47	0.020	0.024	0.058	0.061	0.142	0.651	0.238	−0.001	−0.063	−0.167	−0.056	−0.066	0.309	0.010
		SSNN-47	0.024	0.033	0.081	0.075	0.148	0.639	0.227	0.003	−0.054	−0.144	−0.042	−0.060	0.297	
17	Flavodoxin	Real	0.209	0.108	0.108	0.108	0.264	0.203								
		SELCON3-47	0.136	0.117	0.153	0.090	0.212	0.290	0.274	−0.073	0.009	0.045	−0.018	−0.052	0.087	0.052
		SSNN-47	0.160	0.125	0.108	0.074	0.227	0.306	0.222	−0.049	0.017	0.000	−0.034	−0.037	0.103	
18	γ-Crystalin	Real	0.006	0.086	0.299	0.161	0.109	0.339								
		SELCON3-47	0.003	−0.006	0.237	0.116	0.254	0.392	0.200	−0.003	−0.092	−0.062	−0.045	0.145	0.053	−0.018
		SSNN-47	0.035	0.049	0.303	0.124	0.219	0.270	0.219	0.029	−0.037	0.004	−0.037	0.110	−0.069	
19	Green Fluorescent Protein	Real	0.004	0.064	0.347	0.093	0.191	0.301								
		SELCON3-47	0.022	0.045	0.245	0.118	0.242	0.293	0.181	0.018	−0.019	−0.102	0.025	0.051	−0.008	0.047
		SSNN-47	0.049	0.049	0.306	0.131	0.204	0.261	0.134	0.045	−0.015	−0.041	0.038	0.013	−0.040	
20	Glyceraldehyde-3-phosphate dehydrogenase	Real	0.172	0.102	0.115	0.093	0.217	0.301								
		SELCON3-47	0.157	0.134	0.097	0.078	0.226	0.300	0.080	−0.015	0.032	−0.018	−0.015	0.009	−0.001	−0.002
		SSNN-47	0.168	0.137	0.095	0.079	0.224	0.298	0.081	−0.004	0.035	−0.020	−0.014	0.007	−0.003	
21	Glutathione Reductase	Real	0.188	0.142	0.140	0.096	0.172	0.262								
		SELCON3-47	0.145	0.134	0.102	0.080	0.221	0.316	0.163	−0.043	−0.008	−0.038	−0.016	0.049	0.054	−0.058
		SSNN-47	0.125	0.117	0.115	0.096	0.247	0.300	0.222	−0.063	−0.025	−0.025	0.000	0.075	0.038	
22	Hemoglobin	Real	0.537	0.223	0.000	0.000	0.105	0.136								
		SELCON3-47	0.498	0.228	0.000	0.000	0.167	0.156	0.062	−0.039	0.005	0.000	0.000	0.062	0.020	0.003
		SSNN-47	0.497	0.217	0.013	0.009	0.078	0.185	0.059	−0.040	−0.006	0.013	0.009	−0.027	0.049	
23	Hemerythrin	Real	0.478	0.197	0.000	0.000	0.111	0.215								
		SELCON3-47	0.441	0.198	0.052	0.043	0.064	0.209	0.093	−0.037	0.001	0.052	0.043	−0.047	−0.006	0.018
		SSNN-47	0.447	0.227	0.037	0.027	0.084	0.180	0.075	−0.031	0.030	0.037	0.027	−0.027	−0.035	
24	Rat Intestinal Fatty Acid Binding Protein	Real	0.053	0.061	0.432	0.152	0.152	0.152								
		SELCON3-47	0.117	0.093	0.189	0.094	0.219	0.267	0.685	0.064	0.032	−0.243	−0.058	0.067	0.115	−0.183
		SSNN-47	0.217	0.134	0.101	0.073	0.210	0.264	0.868	0.164	0.073	−0.331	−0.079	0.058	0.112	
25	Insulin	Real	0.294	0.235	0.020	0.040	0.050	0.361								
		SELCON3-47	0.253	0.217	0.072	0.061	0.195	0.199	0.487	−0.041	−0.018	0.052	0.021	0.145	−0.162	0.115
		SSNN-47	0.212	0.154	0.092	0.074	0.175	0.293	0.372	−0.082	−0.081	0.072	0.034	0.125	−0.068	
26	Lactate Dehydrogenase	Real	0.277	0.161	0.088	0.073	0.155	0.246								
		SELCON3-47	0.276	0.178	0.073	0.054	0.158	0.264	0.064	−0.001	0.017	−0.015	−0.019	0.003	0.018	−0.073
		SSNN-47	0.329	0.199	0.063	0.050	0.098	0.262	0.137	0.052	0.038	−0.025	−0.023	−0.057	0.016	
27	Lysozyme	Real	0.202	0.217	0.016	0.047	0.298	0.221								
		SELCON3-47	0.213	0.172	0.060	0.062	0.152	0.325	0.294	0.011	−0.045	0.044	0.015	−0.146	0.104	−0.014
		SSNN-47	0.167	0.129	0.086	0.084	0.218	0.315	0.308	−0.035	−0.088	0.070	0.037	−0.080	0.094	
28	Myoglobin	Real	0.582	0.222	0.000	0.000	0.052	0.144								
		SELCON3-47	0.629	0.247	0.037	−0.001	−0.018	0.142	0.060	0.047	0.025	0.037	−0.001	−0.070	−0.002	−0.073
		SSNN-47	0.462	0.218	0.010	0.012	0.070	0.227	0.134	−0.120	−0.004	0.010	0.012	0.018	0.083	

Table 1. (*Continued*)

Protein number	Protein name	Method	α-regular	α-distorted	β-regular	β-distorted	Turns	Other	Structural NRMSD	Fit minus real: α-regular	Fit minus real: α-distorted	Fit minus real: β-regular	Fit minus real: β-distorted	Fit minus real: turns	Fit minus real: other	SELCON3 NRMSD - SSNN NRMSD
29	Nuclease	Real	0.094	0.101	0.081	0.107	0.289	0.328								
		SELCON3-47	0.175	0.150	0.091	0.067	0.206	0.317	0.217	0.081	0.049	0.010	−0.040	−0.083	−0.011	−0.119
		SSNN-47	0.185	0.148	0.113	0.085	0.197	0.272	0.336	0.091	0.047	0.032	−0.022	−0.092	−0.056	
30	Papain	Real	0.137	0.123	0.094	0.075	0.175	0.396								
		SELCON3-47	0.109	0.091	0.165	0.102	0.229	0.343	0.187	−0.028	−0.032	0.071	0.027	0.054	−0.053	0.071
		SSNN-47	0.080	0.073	0.147	0.088	0.167	0.444	0.116	−0.057	−0.050	0.053	0.013	−0.008	0.048	
31	Parvalbumin	Real	0.278	0.287	0.000	0.037	0.194	0.204								
		SELCON3-47	0.285	0.193	0.031	0.054	0.194	0.266	0.190	0.007	−0.094	0.031	0.017	0.000	0.062	−0.107
		SSNN−47	0.229	0.171	0.064	0.061	0.155	0.321	0.297	−0.049	−0.116	0.064	0.024	−0.039	0.117	
32	Phosphoglycerate Kinase	Real	0.210	0.135	0.043	0.067	0.231	0.313								
		SELCON3-47	0.367	0.222	0.045	0.040	0.107	0.244	0.288	0.157	0.087	0.002	−0.027	−0.124	−0.069	0.010
		SSNN-47	0.276	0.197	0.077	0.060	0.143	0.246	0.278	0.066	0.062	0.034	−0.007	−0.088	−0.067	
33	Pepsinogen	Real	0.051	0.154	0.235	0.151	0.165	0.243								
		SELCON3-47	0.035	0.035	0.256	0.123	0.223	0.316	0.227	−0.016	−0.119	0.021	−0.028	0.058	0.073	0.011
		SSNN-47	0.033	0.053	0.269	0.130	0.228	0.286	0.216	−0.018	−0.101	0.034	−0.021	0.063	0.043	
34	Prealbumin	Real	0.031	0.031	0.307	0.142	0.165	0.323								
		SELCON3-47	0.011	0.082	0.273	0.121	0.214	0.282	0.140	−0.020	0.051	−0.034	−0.021	0.049	−0.041	−0.081
		SSNN-47	0.060	0.093	0.283	0.144	0.192	0.229	0.222	0.029	0.062	−0.024	0.002	0.027	−0.094	
35	Rhodanase	Real	0.150	0.147	0.041	0.068	0.235	0.359								
		SELCON3-47	0.214	0.154	0.092	0.076	0.185	0.284	0.240	0.064	0.007	0.051	0.008	−0.050	−0.075	0.063
		SSNN-47	0.204	0.140	0.086	0.071	0.198	0.301	0.177	0.054	−0.007	0.045	0.003	−0.037	−0.058	
36	Ribonuclease A	Real	0.113	0.097	0.218	0.113	0.218	0.242								
		SELCON3-47	0.114	0.109	0.154	0.102	0.220	0.306	0.184	0.001	0.012	−0.064	−0.011	0.002	0.064	0.049
		SSNN-47	0.083	0.101	0.191	0.098	0.228	0.297	0.135	−0.030	0.004	−0.027	−0.015	0.010	0.055	
37	Substilin BPN	Real	0.171	0.131	0.098	0.080	0.225	0.295								
		SELCON3-47	0.108	0.082	0.136	0.105	0.223	0.325	0.162	−0.063	−0.049	0.038	0.025	−0.002	0.030	0.111
		SSNN-47	0.187	0.138	0.087	0.078	0.207	0.302	0.051	0.016	0.007	−0.011	−0.002	−0.018	0.007	
38	Substilin novo	Real	0.113	0.102	0.065	0.073	0.295	0.353								
		SELCON3-47	0.202	0.139	0.117	0.083	0.199	0.275	0.352	0.089	0.037	0.052	0.010	−0.096	−0.078	0.015
		SSNN-47	0.229	0.161	0.073	0.070	0.165	0.302	0.337	0.116	0.059	0.008	−0.003	−0.130	−0.051	
39	Superoxide Dismutase	Real	0.000	0.018	0.248	0.119	0.298	0.316								
		SELCON3-47	0.060	0.091	0.224	0.122	0.210	0.280	0.253	0.060	0.073	−0.024	0.003	−0.088	−0.036	0.002
		SSNN-47	0.083	0.081	0.177	0.098	0.183	0.378	0.251	0.083	0.063	−0.071	−0.021	−0.115	0.062	
40	T4 Lysozyme	Real	0.421	0.244	0.049	0.037	0.116	0.134								
		SELCON3-47	0.446	0.214	0.033	0.028	0.092	0.175	0.063	0.025	−0.030	−0.016	−0.009	−0.024	0.041	−0.034
		SSNN-47	0.432	0.210	0.028	0.022	0.091	0.217	0.097	0.011	−0.034	−0.021	−0.015	−0.025	0.083	
41	Thermolysin	Real	0.282	0.133	0.070	0.095	0.215	0.206								
		SELCON3-47	0.258	0.156	0.107	0.065	0.148	0.270	0.218	−0.024	0.023	0.037	−0.030	−0.067	0.064	−0.052
		SSNN-47	0.265	0.214	0.045	0.048	0.114	0.315	0.270	−0.018	0.081	−0.025	−0.047	−0.101	0.109	
42	Tumor Necrosis Factor	Real	0.000	0.019	0.293	0.140	0.219	0.329								
		SELCON3-47	0.075	0.122	0.340	0.202	0.091	0.063	0.488	0.075	0.103	0.047	0.062	−0.128	−0.266	0.392
		SSNN-47	0.017	0.055	0.303	0.149	0.189	0.287	0.096	0.017	0.036	0.010	0.009	−0.030	−0.042	
43	Triose Phosphate Isomerase	Real	0.236	0.210	0.090	0.064	0.124	0.276								

Table 1. (*Continued*)

Protein number	Protein name	Method	α-regular	α-distorted	β-regular	β-distorted	Turns	Other	Structural NRMSD	Fit minus real: α-regular	Fit minus real: α-distorted	Fit minus real: β-regular	Fit minus real: β-distorted	Fit minus real: turns	Fit minus real: other	SELCON3 NRMSD - SSNN NRMSD
		SELCON3-47	0.339	0.181	0.060	0.049	0.150	0.230	0.175	0.103	−0.029	−0.030	−0.015	0.026	−0.046	0.012
		SSNN-47	0.331	0.187	0.061	0.052	0.136	0.234	0.163	0.095	−0.023	−0.029	−0.012	0.012	−0.042	
44	Apo-cytochrome C (5°C) denatured	Real	0.020	0.020	0.020	0.020	0.020	0.900								
		SELCON3-47	0.023	0.031	0.053	0.040	0.061	0.754	0.088	0.003	0.011	0.033	0.020	0.041	−0.146	0.012
		SSNN-47	0.028	0.034	0.058	0.040	0.063	0.776	0.076	0.008	0.014	0.038	0.020	0.043	−0.124	
45	Apo-cytochrome C (90°C) denatured	Real	0.020	0.020	0.020	0.020	0.020	0.900								
		SELCON3-47	0.038	0.034	0.075	0.060	0.113	0.673	0.163	0.018	0.014	0.055	0.040	0.093	−0.227	−0.406
		SSNN-47	0.040	0.062	0.175	0.099	0.199	0.426	0.569	0.020	0.042	0.155	0.079	0.179	−0.474	
46	Ribonuclease (20°C) denatured	Real	0.020	0.020	0.020	0.020	0.020	0.900								
		SELCON3-47	0.021	0.059	0.073	0.051	0.100	0.729	0.117	0.001	0.039	0.053	0.031	0.080	−0.171	0.049
		SSNN-47	0.024	0.035	0.055	0.039	0.060	0.788	0.068	0.004	0.015	0.035	0.019	0.040	−0.112	
47	Staphyllococcal Nuclease (6°C) denatured	Real	0.020	0.020	0.020	0.020	0.020	0.900								
		SELCON3-47	0.004	0.025	0.040	0.032	0.064	0.823	0.046	−0.016	0.005	0.020	0.012	0.044	−0.077	−0.033
		SSNN-47	0.028	0.037	0.058	0.040	0.065	0.772	0.079	0.008	0.017	0.038	0.020	0.045	−0.128	
48	Staphyllococcal Nuclease (70°C) denatured	Real	0.020	0.020	0.020	0.020	0.020	0.900								
		SELCON3-47	0.039	0.047	0.098	0.055	0.098	0.647	0.188	0.019	0.027	0.078	0.035	0.078	−0.253	0.026
		SSNN-47	0.035	0.049	0.084	0.057	0.102	0.674	0.162	0.015	0.029	0.064	0.037	0.082	−0.226	
	Sum of absolute values	SELCON3-47							10.56	1.84	1.65	2.78	1.32	2.98	4.49	−0.121
	Sum of absolute values	SSNN-47							10.69	2.17	1.93	2.57	1.28	2.81	4.53	

SSNN was trained with 47 proteins in a LOOCV for 28,000 iterations, a map size of 40 × 40, initial neighbourhood of 20, BMUs = 5, $L_0 = 0.1$, $t_1 = 7{,}000$ iterations, and a $k_1 = 5 \times 10^{-6}$. SELCON3-47 was run using the executable code available at http://lamar.colostate.edu/∼sreeram/CDPro/main.html using input files based on CDDATA.48 and SSDATA.48 but with one spectrum removed to be the test spectrum and structure each time.

trained SOM which in our hands produced clearly poor results with inconsistencies between the model spectrum and the spectral prediction. As we could not find a retrainable executable version of SOMCD, we took the structure predictions/estimations that were available in the SOMCD paper (although we could not reproduce these date with the available code).[18] The proteins tested and reported on in that paper include a subset of 33 proteins from CDDATA.48. Supporting Information, Table S3 contains a comparison of SELCON3-47 and SSNN-47 to the published SOMCD output in the 4 structure format used.[18] In summary, the mean structural NRMSDs of SOMCD, SELCON3-47, SSNN-47 are 0.4 ± 0.1, 0.4 ± 0.3, and 0.4 ± 0.4, with SSNN-47 performing better than SOMCD for 22 of the 33 proteins.

Conclusions

This work has built upon the basis of the concepts of the K2d[12] and SOMCD[18] neural network concepts to make a new SOM neural network, called SSNN or "secondary structure neural network" to determine structural knowledge from CD spectra. SSNN comes in three independent parts: SSNN1 which is the spectral training stage that sorts the reference set spectra on a grid (currently implemented with a 40×40 grids) and interpolates to place a spectrum at each node; SSNN2 which allocates secondary structure vectors that correspond to the spectra to the nodes on a second 40×40 map; and the SSNN3 module which identifies the structure vector of an unknown spectrum and outputs a measure of its accuracy visually by plotting experimental and fitted spectra and numerically with the spectral NRMSD. SSNN is available in a pretrained version that includes wavelengths from 240 to 190 nm from CDPro reference data set CDDATA.48. When comparing the overlay of the best-fit spectrum on the original spectrum and the structure percentage accuracies, SSNN is an improvement on the previous SOM approaches. Further extensions simply require the SSNN1 and SSNN2 modules to be performed with a revised reference set, including structures.

SSNN compares well to the statistical program SELCON3 for protein secondary structure prediction from CD spectra. Overall SELCON3 predicts α-helical and other structures slightly better than SSNN which is better for β-sheets and turns. However, the difference is small. In a comparison of the most commonly used statistical methods, SELCON3, CONTIN, and CDSSTR, Sreerama and Woody found SELCON3 does best for α-regular, β-regular, and turn structures, CONTIN is best for α-distorted and turns, and CDSSTR is best for β-distorted.[19] Sreerama and Woody's overall conclusion was to try all three methods. We would suggest adding SSNN into this process and having confidence in a structure prediction if the methods are in accord and investigate further if they are not. Due to the simple metrics currently used by all fitting programs to assess goodness of spectral fit, we still recommend complementing fitting programs with a visual inspection of the overlay of experimental data and model spectrum, particularly in the 215-nm region of the spectrum. Because of the restricted formats in which K2d and SOMCD are available we do not recommend them.

Most of the statistical protein structure prediction methodologies occasionally make structure predictions of negative numbers, for example, they will predict that a protein has -1.8% turn structure (unless a check is put on them). This is, of course, physically impossible, and happens because the methodologies make extrapolations from the structures present in their reference sets (reference spectra have no negative numbers in them as they are from real proteins). Due to the fact that SSNN only makes interpolations between protein spectra and structures, it can never predict negative structure percentages. However, with SSNN, when the BMUs of the protein are on the edge of the map, the structure prediction may not be very good. Other warning signs are large spectral NRMSDs or the situation where the NRMSD is reasonable, but either the experimental or model spectrum has a negative maximum in the 215-nm region and the other does not (i.e., the β-sheet/α-helix identity).

The agreement between SSNN-47 and SELCON3-47 on the structure of proteins 44–48 in the CDDATA48 database (i.e. the denatured proteins) yet their similar differences from the somewhat arbitrarily assigned (0.02, 0.02, 0.02, 0.02, 0.02, 0.9) suggests that the structure vectors for these five proteins should be replaced by an average of the SSNN-47 and SELCON3-47 structures in Table 3.

Table 2. Structural NRMSDs for SSNN-47 and SELCON-47 from the data in Table 1 for different structural classes of protein.

Program	Overall	>50% α-helix	30–50% α-helix	>30% β-sheet	>50% Other
SSNN-47	0.2 ± 0.2	0.2 ± 0.1	0.2 ± 0.2	0.2 ± 0.2	0.2 ± 0.1
SELCON3–47	0.2 ± 0.2	0.1 ± 0.1	0.3 ± 0.3	0.3 ± 0.2	0.2 ± 0.2
K2d	0.3 ± 0.2	0.1 ± 0.2	0.2 ± 0.1	0.3 ± 0.2	0.3 ± 0.3

"Errors" are one standard deviation of the variation between proteins in the class.

Table 3. Structure NRMSDs and sum of absolute values of the model spectrum structure minus the real value for the three structural categories α-helix, β-sheet, and Other for SSNN-47 and SELCON3-47 for all proteins in CDDATA.48 using data from Table 1 and for K2d using its available format on Dichroweb but testing on CDDATA.48 (some of the test spectra are in the K2d training set).

Method	Sum of structural NRMSDs	Fit minus real: α	Fit minus real: β	Fit minus real: Other
SELCON3-47	11.05	3.37	4.09	4.75
SSNN-47	10.67	3.68	3.60	4.21
K2d	12.32	2.78	4.65	4.75

Acknowledgments

This work has been greatly improved by helpful discussions of the manuscript with R.W. Woody and by discussions of comparisons between methods with P.M. Rodger.

Keywords: artificial neural networks · circular dichroism · intelligent systems · kohonen · protein structure fitting · self organising maps

How to cite this article: V. Hall, A. Nash, E. Hines, A. Rodger. *J. Comput. Chem.* **2013**, DOI: 10.1002/jcc.23456

Additional Supporting Information may be found in the online version of this article.

[1] B. Nordén, A. Rodger, T. R. Dafforn, *Linear Dichroism and Circular Dichroism: A Textbook on Polarized Spectroscopy*; Royal Society of Chemistry: Cambridge, **2010**; 304 p.
[2] A. M. Tsai, J. H. van Zanten, M. Betenbaugh, *J. Biotechnol. Bioeng.* **1998**, *59*, 273.
[3] N. Greenfield, *J. Nat. Protoc.* **2006**, *1*, 2527.
[4] K. Takeda, A. Shigemura, S. Hamada, W. Gu, D. Fang, K. Sasa, K. Hachiya, *J. Protein Chem.* **1992**, *11*, 187.
[5] J. -Y. Cherng, H. Telsma, D. J. A. Crommelin, *W. E. Hennink. Pharma Res.* **1999**, *16*, 1417.
[6] L. Whitmore, B. A. Wallace, *Nucleic Acids Res.* **2004**, *32*, W668.
[7] R. W. Woody, *J. Am. Chem. Soc.* **2009**, *131*, 8234.
[8] D. Waldron, R. Marrington, M. Grant, M. Hicks, A. Rodger, *Chirality* **2010**, *22*, E136.
[9] S. W. Provencher, *Comput. Phys. Commun.* **1978**, *27*, 229.
[10] W. C. Johnson, *Proteins Struct. Funct. Genet.* **1999**, *35*, 307.
[11] N. Sreerama, R. W. Woody, *Anal. Biochem.* **1993**, *209*, 32.
[12] M. A. Andrade, P. Chacon, J. J. Merelo, F. Moran, *Protein Eng.* **1993**, *6*, 383.
[13] A. Lobley, L. Whitmore, B. A. Wallace, *Bioinformatics* **2001**, *18*, 211.
[14] T. Kohonen, *Biol. Cybern.* **1982**, *43*, 59.
[15] C. Louis-Jeune, M. A. Andrade-Navarro, C. Perez-Iratxeta, Available at: http://www.ogic.ca/projects/k2d3/, accessed on 29th september **2013**.
[16] C. Louis-Jeune, M. A. Andrade-Navarro, *C. Perez-Iratxeta Proteins Struct. Funct. Bioinf.* **2012**, *80*, 374.
[17] B. M. Bulheller, *J. D. Hirst. Bioinformatics* **2009**, *25*, 539.
[18] P. Unneberg, J. J. Merelo, P. Chaco, F. Moran, *Proteins Struct. Funct. Genet.* **2001**, *42*, 460.
[19] N. Sreerama, R. W. Woody, *Anal. Biochem.* **2000**, *287*, 252.
[20] G. C. Cawley, *N. L. C. Talbot. Neural Netw.* **2004**, *17*, 1467.
[21] S. M. Kelly, T. J. Jess, N. C. Price, *Biochim. Biophys. Acta* **2005**, *1751*, 119.

Received: 8 August 2013
Accepted: 2 September 2013
Published online on

Elucidating Protein Secondary Structure with Circular Dichroism and a Neural Network

VINCENT HALL, ANTHONY NASH, EVOR HINES, ALISON RODGER

Supplementary information

SI 1 SSNN parameter optimisation

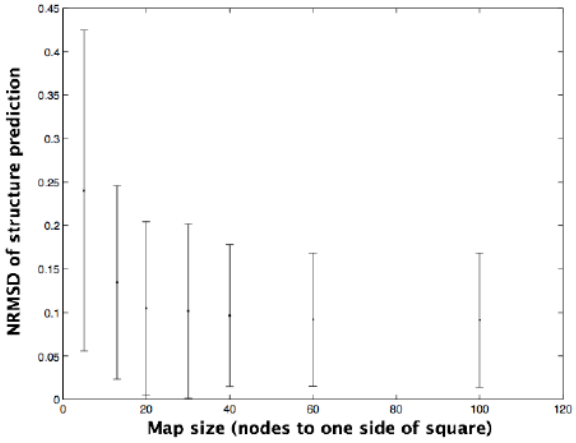


Figure SI1. Dependence of mean spectral NRMSDs of SSNN-47 as a function of map size in SSNN3. "Error bars" are one standard deviation of the variation between proteins. SSNN was trained with 47 proteins in a LOOCV for 20,000 iterations, a map size of 7×7 to 100×100, initial neighbourhood of 40, initial neighbourhood radius = 10, SSNN2 BMUs = 5, SSNN3 BMUs = 40, $L_0 = 0.1$, $t_1 = 7{,}000$ iterations, and a $k_1 = 5\times10^{-6}$.

Table SI1. Dependence of mean structural NRMSDs of SSNN-47 on wavelength range. "Errors" are one standard deviation of the variation between proteins. SSNN was trained with 47 proteins in a LOOCV for 28,000 iterations, a map size of 20×20, initial neighbourhood size = 10, BMUs = 5, $L_0 = 0.1$, $t_1 = 7{,}000$ iterations, and a $k_1 = 5\times10^{-6}$.

Range/nm	Wavelengths/nm	NRMSD
41	240–200	0.14±0.08
46	240–195	0.13±0.08
51	240–190	0.1±0.1

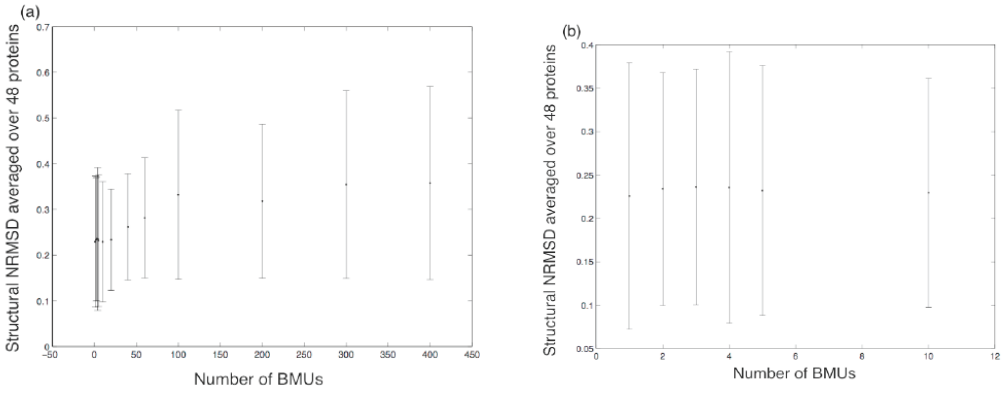


Figure SI2. Dependence of mean structural NRMSDs of SSNN-47 on number of BMUs in SSNN3.

(a) full range of BMUs tested, (b) small numbers of BMUs. "Error bars" are one standard deviation of the variation between proteins. SSNN was trained with 47 proteins in a LOOCV for 20,000 iterations, a map size of 13×13 for 1–100 BMUs, 20 for 200–300 BMUs, 22 for 400 BMUs (to ensure the map has enough nodes), initial neighbourhood of 10, SSNN2 BMUs = 5, $L_0 = 0.1$, t_1 = 7,000 iterations, and a $k_1 = 5\times10^{-6}$.

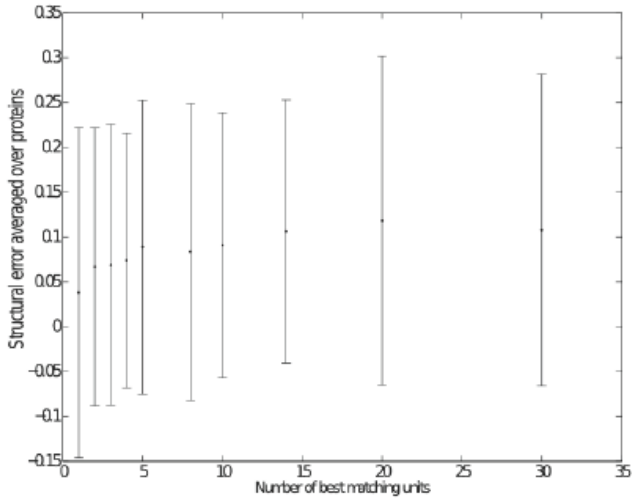


Figure SI3. Dependence of mean structural NRMSDs of SSNN-23 on number of BMUs in SSNN3. "Error bars" are one standard deviation of the variation between proteins. This was to see what effect changing the size of the reference set had on the optimum number of BMUs. SSNN was trained with 23 proteins in a LOOCV for 5,000 iterations, a map size of 40×40, initial neighbourhood of 20, SSNN2 BMUs = 5, $L_0 = 0.1$, $t_1 = 7{,}000$ iterations, and a $k_1 = 5\times10^{-6}$.

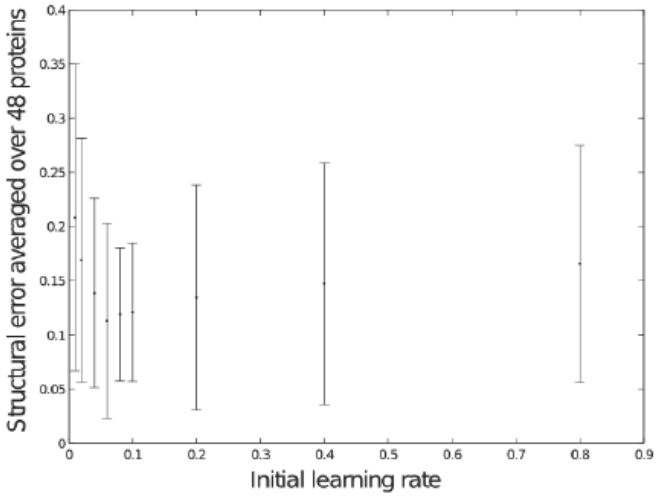


Figure SI4. Dependence of mean structural NRMSDs of SSNN-47 on learning rate. "Error bars" are one standard deviation of the variation between proteins. SSNN was trained with 47 proteins in a LOOCV for 1,000 iterations, a map size of 40×40, initial neighbourhood of 10, BMUs = 5, t_1 = 7,000 iterations, and a $k_1 = 5\times10^{-6}$.

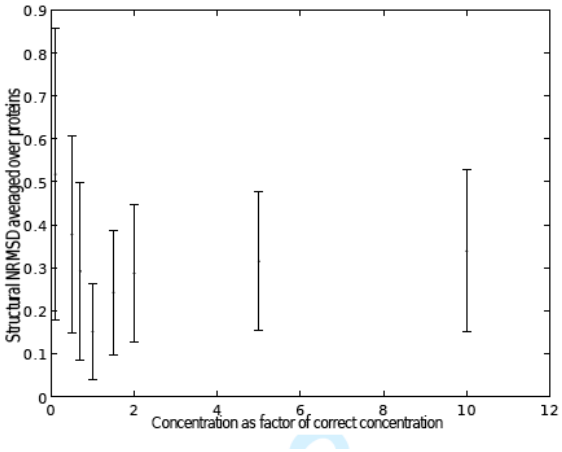


Figure SI5. SSNN mean structural NRMSDs for SSNN-47 as a function of protein concentration error. "Error bars" are one standard deviation of the variation between proteins. Position 1 on the x-axis of this graph is the correct concentration. SSNN was trained with 47 proteins in a LOOCV for 1,000 iterations, a map size of 40×40, initial neighbourhood of 10, BMUs = 5, $L_0 = 0.06$, $t_1 = 7{,}000$ iterations, and a $k_1 = 5\times10^{-6}$.

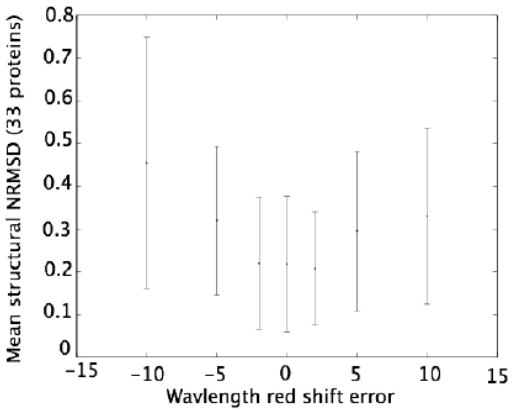


Figure SI6. Spectral NRMSD versus wavelength calibration error for SSNN. –10 nm denotes at 10 nm red (*i.e.* to shorter wavelength), +10 nm is a 10 nm blue-shift *etc*. "Error bars" are one standard deviation of the variation between proteins. SSNN was trained with 33 proteins in a LOOCV for 1000 iterations using the leave-one-out method, a map size of 13×13, initial neighbourhood size = 10, 5 BMUs, an $L_0 = 0.1$, a t_1 of 7,000 iterations, and a $k_1 = 5\times10^{-6}$.

Table SI2. Prediction of secondary structure content in terms of the three categories used by K2d by SELCON3-47, SSNN-47, as outlined in the main paper compared, with K2d performed using Dichroweb [1] and a training reference set that uses some of the test proteins. The final columns contain the difference between the models and the real data together with a sum of the absolute values of the differences.

Protein number	Protein name	Method	α	β	Other	Structural NRMSD	Fit minus real: α	Fit minus real: β	Fit minus real: other
1	α-Bungarotoxin	Real	0.000	0.109	0.892				
		K2d	0.11	0.4	0.48	0.806	0.110	0.291	-0.412
		SELCON3-47	0.035	0.428	0.543	0.539	0.035	0.319	-0.349
		SSNN-47	0.073	0.332	0.594	0.128	0.073	0.223	-0.298
2	Alcohol Dehydrogenase	Real	0.254	0.235	0.511				
		K2d	0.33	0.19	0.49	0.175	0.076	-0.045	-0.021
		SELCON3-47	0.241	0.209	0.528	0.061	-0.013	-0.026	0.017
		SSNN-47	0.260	0.274	0.466	0.258	0.006	0.039	-0.045
3	Adenylate Kinase	Real	0.546	0.129	0.325				
		K2d	0.54	0.12	0.34	0.025	-0.006	-0.009	0.015
		SELCON3-47	0.419	0.142	0.436	0.332	-0.127	0.013	0.111
		SSNN-47	0.404	0.145	0.450	0.039	-0.142	0.016	0.125
4	Azurin	Real	0.109	0.250	0.640				
		K2d	0.16	0.36	0.48	0.362	0.051	0.110	-0.160
		SELCON3-47	0.234	0.259	0.518	0.356	0.125	0.009	-0.122
		SSNN-47	0.252	0.238	0.510	0.060	0.143	-0.012	-0.130
5	β-lactoglobulin	Real	0.167	0.410	0.423				
		K2d	0.3	0.15	0.55	0.460	0.133	-0.260	0.127
		SELCON3-47	0.182	0.271	0.533	0.293	0.015	-0.139	0.110
		SSNN-47	0.208	0.306	0.486	0.134	0.041	-0.104	0.063
6	Bence Jones Protein	Real	0.028	0.490	0.481				
		K2d	0.03	0.5	0.47	0.018	0.002	0.010	-0.011
		SELCON3-47	0.003	0.281	0.739	0.261	-0.025	-0.209	0.258
		SSNN-47	0.058	0.450	0.492	0.404	0.030	-0.040	0.011
7	Bovine Pancreatic Trypsin Inhibitor	Real	0.207	0.241	0.552				
		K2d	0.28	0.33	0.39	1.043	0.073	0.089	-0.162
		SELCON3-47	0.192	0.252	0.539	0.038	-0.015	0.011	-0.013
		SSNN-47	0.123	0.280	0.597	0.115	-0.084	0.039	0.045
8	Carbonic Anhydrase	Real	0.162	0.286	0.552				
		K2d	0.08	0.44	0.48	0.272	-0.082	0.154	-0.072
		SELCON3-47	0.075	0.174	0.728	0.200	-0.087	-0.112	0.176
		SSNN-47	0.060	0.098	0.843	0.102	-0.102	-0.188	0.291
9	CGA	Real	0.135	0.320	0.545				
		K2d	0.19	0.3	0.51	0.123	0.055	-0.020	-0.035
		SELCON3-47	0.121	0.156	0.724	0.233	-0.014	-0.164	0.179
		SSNN-47	0.115	0.126	0.759	0.042	-0.020	-0.194	0.214
10	α-Chymotrypsin	Real	0.114	0.314	0.571				
		K2d	0.09	0.35	0.56	0.055	-0.024	0.036	-0.011
		SELCON3-47	0.069	0.226	0.686	0.142	-0.045	-0.088	0.115
		SSNN-47	0.107	0.273	0.620	0.100	-0.007	-0.041	0.049
11	Colicin A	Real	0.754	0.000	0.246				
		K2d	0.78	0	0.21	0.033	0.026	0.000	-0.036
		SELCON3-47	0.694	0.007	0.318	0.079	-0.060	0.007	0.072
		SSNN-47	0.700	0.030	0.271	0.046	-0.054	0.030	0.025
12	Concanavalin A	Real	0.038	0.464	0.498				
		K2d	0.02	0.51	0.47	0.067	-0.018	0.046	-0.028
		SELCON3-47	0.088	0.374	0.524	0.141	0.050	-0.090	0.026
		SSNN-47	0.065	0.456	0.479	0.134	0.027	-0.008	-0.019
13	Carboxypepsidase A	Real	0.381	0.163	0.456				
		K2d	0.37	0.15	0.48	0.051	-0.011	-0.013	0.024
	Cytochrome C	SELCON3-47	0.233	0.361	0.402	0.864	-0.148	0.198	-0.054
		SSNN-47	0.168	0.380	0.452	0.171	-0.213	0.217	-0.004
14		Real	0.408	0.000	0.592				
		K2d	0.38	0.05	0.57	0.068	-0.028	0.050	-0.022
		SELCON3-47	0.334	0.174	0.501	0.370	-0.074	0.174	-0.091
		SSNN-47	0.295	0.190	0.516	0.079	-0.113	0.190	-0.076
15	EcoR1 Endonuclease	Real	0.319	0.178	0.503				
		K2d	0.3	0.15	0.55	0.084	-0.019	-0.028	0.047
		SELCON3-47	0.327	0.170	0.499	0.021	0.008	-0.008	-0.004
		SSNN-47	0.371	0.112	0.518	0.107	0.052	-0.066	0.015
16	Elastase	Real	0.108	0.342	0.550				
		K2d	0.07	0.51	0.42	0.283	-0.038	0.168	-0.130
		SELCON3-47	0.044	0.119	0.793	0.259	-0.064	-0.223	0.243
		SSNN-47	0.057	0.156	0.786	0.032	-0.051	-0.186	0.236
17	Flavodoxin	Real	0.317	0.216	0.467				
		K2d	0.36	0.14	0.5	0.150	0.043	-0.076	0.033
		SELCON3-47	0.253	0.243	0.502	0.173	-0.064	0.027	0.035
		SSNN-47	0.285	0.182	0.533	0.124	-0.032	-0.034	0.066
18	γ-Crystalin	Real	0.092	0.460	0.448				
		K2d	0.02	0.51	0.47	0.106	-0.072	0.050	0.022
		SELCON3-47	-0.003	0.353	0.646	0.217	-0.095	-0.107	0.198
		SSNN-47	0.084	0.427	0.489	0.278	-0.008	-0.033	0.041
19	Green Fluorescent Protein	Real	0.068	0.440	0.492				
		K2d	0.05	0.48	0.47	0.066	-0.018	0.040	-0.022
		SELCON3-47	0.067	0.363	0.535	0.109	-0.001	-0.077	0.043
		SSNN-47	0.098	0.438	0.465	0.169	0.030	-0.002	-0.027
20	Glyceraldehyde-3-phosphate dehydrogenase	Real	0.274	0.208	0.518				
		K2d	0.3	0.12	0.58	0.139	0.026	-0.088	0.062
		SELCON3-47	0.291	0.175	0.526	0.062	0.017	-0.033	0.008
		SSNN-47	0.305	0.174	0.521	0.024	0.031	-0.034	0.003
21	Glutathione Reductase	Real	0.330	0.236	0.434				
		K2d	0.25	0.16	0.59	0.257	-0.080	-0.076	0.156
		SELCON3-47	0.279	0.182	0.537	0.207	-0.051	-0.054	0.103
		SSNN-47	0.242	0.211	0.547	0.083	-0.088	-0.025	0.113
22	Hemoglobin	Real	0.760	0.000	0.241				
		K2d	0.73	0.06	0.21	0.064	-0.030	0.060	-0.031
		SELCON3-47	0.726	0.000	0.323	0.071	-0.034	0.000	0.082
		SSNN-47	0.660	0.032	0.308	0.068	-0.100	0.032	0.067
23	Hemerythrin	Real	0.675	0.000	0.326				
		K2d	0.61	0.07	0.32	0.102	-0.065	0.070	-0.006
		SELCON3-47	0.639	0.095	0.273	0.122	-0.036	0.095	-0.053
		SSNN-47	0.673	0.063	0.263	0.045	-0.002	0.063	-0.063
24	Rat Intestinal Fatty Acid Binding Protein	Real	0.114	0.584	0.304				
		K2d	0.35	0.16	0.49	0.909	0.236	-0.424	0.186
		SELCON3-47	0.210	0.283	0.486	0.763	0.096	-0.301	0.182
		SSNN-47	0.351	0.175	0.474	0.345	0.237	-0.409	0.170

Protein number	Protein name	Method	α	β	Other	Structural NRMSD	Fit minus real: α	Fit minus real: β	Fit minus real: other
25	Insulin	Real	0.529	0.060	0.411				
		K2d	0.51	0.24	0.25	0.518	-0.019	0.180	-0.161
		SELCON3-47	0.470	0.133	0.394	0.163	-0.059	0.073	-0.017
		SSNN-47	0.366	0.166	0.468	0.252	-0.163	0.106	0.057
26	Lactate Dehydrogenase	Real	0.438	0.161	0.401				
		K2d	0.46	0.23	0.31	0.292	0.022	0.069	-0.091
		SELCON3-47	0.454	0.127	0.422	0.076	0.016	-0.034	0.021
		SSNN-47	0.528	0.113	0.360	0.135	0.090	-0.048	-0.041
27	Lysozyme	Real	0.419	0.063	0.519				
		K2d	0.4	0.16	0.44	0.261	-0.019	0.097	-0.079
		SELCON3-47	0.385	0.122	0.477	0.130	-0.034	0.059	-0.042
		SSNN-47	0.296	0.170	0.533	0.184	-0.123	0.107	0.014
28	Myoglobin	Real	0.804	0.000	0.196				
		K2d	0.84	0	0.16	0.035	0.036	0.000	-0.036
		SELCON3-47	0.876	0.036	0.124	0.074	0.072	0.036	-0.072
		SSNN-47	0.680	0.023	0.298	0.230	-0.124	0.023	0.102
29	Nuclease	Real	0.195	0.188	0.617				
		K2d	0.24	0.15	0.61	0.074	0.045	-0.038	-0.007
		SELCON3-47	0.325	0.158	0.523	0.258	0.130	-0.030	-0.094
		SSNN-47	0.333	0.198	0.469	0.145	0.138	0.010	-0.148
30	Papain	Real	0.260	0.169	0.571				
		K2d	0.26	0.15	0.59	0.035	0.000	-0.019	0.019
		SELCON3-47	0.200	0.267	0.572	0.178	-0.060	0.098	0.001
		SSNN-47	0.153	0.235	0.611	0.087	-0.107	0.066	0.040
31	Parvalbumin	Real	0.565	0.037	0.398				
		K2d	0.62	0.05	0.33	0.090	0.055	0.013	-0.068
		SELCON3-47	0.478	0.085	0.460	0.172	-0.087	0.048	0.062
		SSNN-47	0.399	0.125	0.476	0.148	-0.166	0.088	0.078
32	Phosphoglycerate Kinase	Real	0.345	0.110	0.544				
		K2d	0.7	0.03	0.27	0.393	0.355	-0.080	-0.274
		SELCON3-47	0.589	0.085	0.351	0.358	0.244	-0.025	-0.193
		SSNN-47	0.473	0.138	0.389	0.229	0.128	0.028	-0.155
33	Pepsinogen	Real	0.205	0.386	0.408				
		K2d	0.21	0.31	0.48	0.224	0.005	-0.076	0.072
		SELCON3-47	0.070	0.379	0.539	0.232	-0.135	-0.007	0.131
		SSNN-47	0.086	0.399	0.515	0.048	-0.119	0.013	0.107
34	Prealbumin	Real	0.062	0.449	0.488				
		K2d	0.21	0.31	0.48	0.435	0.148	-0.139	-0.008
		SELCON3-47	0.093	0.394	0.496	0.091	0.031	-0.055	0.008
		SSNN-47	0.152	0.427	0.420	0.214	0.090	-0.022	-0.068
35	Rhodanase	Real	0.297	0.109	0.594				
		K2d	0.36	0.22	0.41	0.680	0.063	0.111	-0.184
		SELCON3-47	0.368	0.168	0.469	0.298	0.071	0.059	-0.125
		SSNN-47	0.345	0.157	0.498	0.066	0.048	0.048	-0.096
36	Ribonuclease A	Real	0.210	0.331	0.460				
		K2d	0.23	0.4	0.36	0.418	0.020	0.069	-0.100
		SELCON3-47	0.223	0.256	0.526	0.192	0.013	-0.075	0.066
		SSNN-47	0.184	0.290	0.526	0.087	-0.026	-0.041	0.066
37	Substilin BPN	Real	0.302	0.178	0.520				
		K2d	0.31	0.11	0.58	0.112	0.008	-0.068	0.060
		SELCON3-47	0.190	0.241	0.548	0.212	-0.112	0.063	0.028
		SSNN-47	0.325	0.165	0.509	0.268	0.023	-0.013	-0.011
38	Substilin novo	Real	0.215	0.138	0.648				
		K2d	0.38	0.09	0.53	0.274	0.165	-0.048	-0.118
		SELCON3-47	0.341	0.200	0.474	0.471	0.126	0.062	-0.174
		SSNN-47	0.390	0.143	0.467	0.135	0.175	0.005	-0.181
39	Superoxide Dismutase	Real	0.018	0.367	0.614				
		K2d	0.14	0.33	0.53	0.226	0.122	-0.037	-0.084
		SELCON3-47	0.151	0.346	0.490	0.312	0.133	-0.021	-0.124
		SSNN-47	0.164	0.276	0.561	0.146	0.146	-0.091	-0.053
40	T4 Lysozyme	Real	0.665	0.086	0.250				
		K2d	0.62	0.08	0.3	0.072	-0.045	-0.006	0.050
		SELCON3-47	0.660	0.061	0.267	0.030	-0.005	-0.025	0.017
		SSNN-47	0.642	0.050	0.308	0.045	-0.023	-0.036	0.058
41	Thermolysin	Real	0.415	0.165	0.421				
		K2d	0.49	0.18	0.33	0.221	0.075	0.015	-0.091
		SELCON3-47	0.414	0.172	0.418	0.018	-0.001	0.007	-0.003
		SSNN-47	0.478	0.093	0.429	0.154	0.063	-0.072	0.008
42	Tumor Necrosis Factor	Real	0.019	0.433	0.548				
		K2d	0.03	0.5	0.47	0.127	0.011	0.067	-0.078
		SELCON3-47	0.197	0.542	0.154	0.663	0.178	0.109	-0.394
		SSNN-47	0.072	0.452	0.476	0.510	0.053	0.019	-0.072
43	Triose Phosphate Isomerase	Real	0.446	0.154	0.400				
		K2d	0.6	0.07	0.33	0.206	0.154	-0.084	-0.070
		K2d	0.03	0.5	0.47	0.894	-0.570	0.430	0.140
		SSNN-47	0.518	0.112	0.370	0.899	0.072	-0.042	-0.030
44	Apo-cytochrome C (5°C) denatured	Real	0.040	0.040	0.920				
		K2d	0.04	0.33	0.63	0.401	0.000	0.290	-0.290
		SELCON3-47	0.054	0.093	0.815	0.090	0.014	0.053	-0.105
		SSNN-47	0.062	0.098	0.839	0.020	0.022	0.058	-0.081
45	Apo-cytochrome C (90°C) denatured	Real	0.040	0.040	0.920				
		K2d	0.08	0.41	0.5	0.771	0.040	0.370	-0.420
		SELCON3-47	0.072	0.135	0.786	0.135	0.032	0.095	-0.134
		SSNN-47	0.102	0.274	0.625	0.237	0.062	0.234	-0.296
46	Ribonuclease (20°C) denatured	Real	0.040	0.040	0.920				
		K2d	0.06	0.23	0.72	0.242	0.020	0.190	-0.200
		SELCON3-47	0.080	0.124	0.829	0.100	0.040	0.084	-0.091
		SSNN-47	0.059	0.094	0.848	0.030	0.019	0.054	-0.072
47	Staphyllococcal Nuclease (6°C) denatured	Real	0.040	0.040	0.920				
		K2d	0.02	0.12	0.86	0.070	-0.020	0.080	-0.060
		SELCON3-47	0.029	0.072	0.887	0.032	-0.011	0.032	-0.033
		SSNN-47	0.065	0.098	0.837	0.050	0.025	0.058	-0.083
48	Staphyllococcal Nuclease (70°C) denatured	Real	0.040	0.040	0.920				
		K2d	0.05	0.33	0.62	0.423	0.010	0.290	-0.300
		SELCON3-47	0.086	0.153	0.745	0.187	0.046	0.113	-0.175
		SSNN-47	0.084	0.141	0.776	0.028	0.044	0.101	-0.144
	Sum of absolute values	K2d				12.32	2.78	4.65	4.75
	Sum of absolute values	SELCON3-47				11.28	3.52	4.18	4.89
	Sum of absolute values	SSNN				7.43	3.73	3.61	4.26

Table SI3. Structure vectors and structure NRMSDs, differences from true fit of SOMCD, SELCON3-47 and SSNN-47 in 4 structure-type format.

Protein number	Protein name	Method	α-regular	β-regular	Turns	Other	Structural NRMSD	Fit minus real: α-regular	Fit minus real: β-regular	Fit minus real: turns	Fit minus real: other	SOMCD NRMSD – SSNN NRMSD
1	Alcohol	Real	0.254	0.235	0.214	0.297						
	Dehydrogenase	SOMCD	0.400	0.220	0.120	0.260	0.438	0.146	-0.015	-0.094	-0.037	-0.232
		SELCON3-47	0.241	0.209	0.231	0.297	0.191	-0.013	-0.026	0.017	0.000	
		SSNN-47	0.260	0.274	0.224	0.242	0.670	0.006	0.039	0.010	-0.055	
2	Azurin	Real	0.109	0.250	0.312	0.328						
		SOMCD	0.350	0.230	0.110	0.310	0.768	0.241	-0.020	-0.202	-0.018	-0.917
		SELCON3-47	0.234	0.259	0.230	0.288	1.336	0.125	0.009	-0.082	-0.040	
		SSNN-47	0.252	0.238	0.229	0.280	1.685	0.143	-0.012	-0.083	-0.048	
3	β-lactoglobulin	Real	0.167	0.410	0.216	0.207						
		SOMCD	0.220	0.380	0.110	0.290	0.384	0.053	-0.030	-0.106	0.083	-0.280
		SELCON3-47	0.182	0.271	0.212	0.321	0.649	0.015	-0.139	-0.004	0.114	
		SSNN-47	0.208	0.306	0.212	0.274	0.663	0.041	-0.104	-0.004	0.067	
4	Bence Jones Protein	Real	0.028	0.490	0.229	0.252						
		SOMCD	0.100	0.470	0.090	0.340	0.328	0.072	-0.020	-0.139	0.088	0.237
		SELCON3-47	0.003	0.281	0.200	0.539	0.333	-0.025	-0.209	-0.029	0.287	
		SSNN-47	0.058	0.450	0.199	0.293	0.091	0.030	-0.040	-0.030	0.041	
5	Carbonic Anhydrase	Real	0.162	0.286	0.240	0.312						
		SOMCD	0.150	0.290	0.170	0.400	0.396	-0.012	0.004	-0.070	0.088	0.018
		SELCON3-47	0.075	0.174	0.109	0.619	0.333	-0.087	-0.112	-0.131	0.307	
		SSNN-47	0.060	0.098	0.059	0.784	0.379	-0.102	-0.188	-0.181	0.472	
6	α-Chymotrypsin	Real	0.114	0.314	0.200	0.371						
		SOMCD	0.150	0.330	0.130	0.390	0.289	0.036	0.016	-0.070	0.019	0.140
		SELCON3-47	0.069	0.226	0.142	0.544	0.218	-0.045	-0.088	-0.058	0.173	
		SSNN-47	0.107	0.273	0.161	0.459	0.149	-0.007	-0.041	-0.039	0.088	
7	Concanavalin A	Real	0.038	0.464	0.236	0.262						
		SOMCD	0.110	0.430	0.100	0.370	0.389	0.072	-0.034	-0.136	0.108	0.293
		SELCON3-47	0.088	0.374	0.209	0.315	0.208	0.050	-0.090	-0.027	0.053	
		SSNN-47	0.065	0.456	0.178	0.301	0.096	0.027	-0.008	-0.058	0.039	
8	Carboxypepsidase A	Real	0.381	0.163	0.212	0.244						
		SOMCD	0.390	0.230	0.110	0.270	0.333	0.009	0.067	-0.102	0.026	-0.385
		SELCON3-47	0.233	0.361	0.173	0.229	0.667	-0.148	0.198	-0.039	-0.015	
		SSNN-47	0.168	0.380	0.212	0.239	0.718	-0.213	0.217	0.000	-0.005	
9	Cytochrome C	Real	0.408	0.000	0.233	0.359						
		SOMCD	0.390	0.240	0.110	0.270	0.618	-0.018	0.240	-0.123	-0.089	-0.483
		SELCON3-47	0.334	0.174	0.228	0.273	0.649	-0.074	0.174	-0.005	-0.086	
		SSNN-47	0.295	0.190	0.224	0.292	1.101	-0.113	0.190	-0.009	-0.067	
10	EcoR1 Endonuclease	Real	0.319	0.178	0.210	0.293						
		SOMCD	0.460	0.130	0.160	0.260	0.403	0.141	-0.048	-0.050	-0.033	0.213
		SELCON3-47	0.327	0.170	0.202	0.297	0.046	0.008	-0.008	-0.008	0.004	
		SSNN-47	0.371	0.112	0.253	0.265	0.190	0.052	-0.066	0.043	-0.028	
11	Elastase	Real	0.108	0.342	0.208	0.342						
		SOMCD	0.170	0.310	0.120	0.410	0.346	0.062	-0.032	-0.088	0.068	0.038
		SELCON3-47	0.044	0.119	0.142	0.651	0.323	-0.064	-0.223	-0.066	0.309	
		SSNN-47	0.057	0.156	0.148	0.639	0.308	-0.051	-0.186	-0.060	0.297	
12	Flavodoxin	Real	0.317	0.216	0.264	0.203						
		SOMCD	0.370	0.190	0.120	0.320	0.509	0.053	-0.026	-0.144	0.117	0.029
		SELCON3-47	0.253	0.243	0.212	0.290	0.788	-0.064	0.027	-0.052	0.087	
		SSNN-47	0.285	0.182	0.227	0.306	0.480	-0.032	-0.034	-0.037	0.103	
13	γ-Crystalin	Real	0.092	0.460	0.109	0.339						
		SOMCD	0.060	0.450	0.080	0.410	0.187	-0.032	-0.010	-0.029	0.071	-0.009
		SELCON3-47	-0.003	0.353	0.254	0.392	0.266	-0.095	-0.107	0.145	0.053	
		SSNN-47	0.084	0.427	0.219	0.270	0.196	-0.008	-0.033	0.110	-0.069	
14	Glyceraldehyde-3	Real	0.274	0.208	0.217	0.301						
	phosphate	SOMCD	0.410	0.160	0.130	0.300	0.431	0.136	-0.048	-0.087	-0.001	-0.671
	dehydrogenase	SELCON3-47	0.291	0.175	0.226	0.300	0.153	0.017	-0.033	0.009	-0.001	
		SSNN-47	0.295	0.190	0.224	0.292	1.101	-0.113	0.190	-0.009	-0.067	
15	Glutathione Reductase	Real	0.330	0.236	0.172	0.262						
		SOMCD	0.340	0.260	0.120	0.270	0.253	0.010	0.024	-0.052	0.008	-0.449
		SELCON3-47	0.279	0.182	0.221	0.316	0.388	-0.051	-0.054	0.049	0.054	
		SSNN-47	0.242	0.211	0.247	0.300	0.702	-0.088	-0.025	0.075	0.038	
16	Hemoglobin	Real	0.760	0.000	0.105	0.136						
		SOMCD	0.800	0.030	0.060	0.120	0.105	0.040	0.030	-0.045	-0.016	0.004
		SELCON3-47	0.726	0.000	0.167	0.156	0.051	-0.034	0.000	0.062	0.020	
		SSNN-47	0.660	0.032	0.102	0.206	0.100	-0.100	0.032	-0.003	0.070	
17	Hemerythrin	Real	0.675	0.000	0.111	0.215						
		SOMCD	0.750	0.080	0.060	0.120	0.171	0.075	0.080	-0.051	-0.095	0.108
		SELCON3-47	0.639	0.095	0.064	0.209	0.097	-0.036	0.095	-0.047	-0.006	
		SSNN-47	0.673	0.063	0.084	0.180	0.064	-0.002	0.063	-0.027	-0.035	
18	Lactate Dehydrogenase	Real	0.438	0.161	0.155	0.246						
		SOMCD	0.510	0.150	0.070	0.270	0.200	0.072	-0.011	-0.085	0.024	0.063
		SELCON3-47	0.454	0.127	0.158	0.264	0.064	0.016	-0.034	0.003	0.018	
		SSNN-47	0.528	0.113	0.098	0.262	0.137	0.090	-0.048	-0.057	0.016	
19	Lysozyme	Real	0.419	0.063	0.298	0.221						
		SOMCD	0.380	0.200	0.120	0.300	0.584	-0.039	0.137	-0.178	0.079	-0.120
		SELCON3-47	0.385	0.122	0.152	0.325	0.365	-0.034	0.059	-0.146	0.104	
		SSNN-47	0.296	0.170	0.218	0.315	0.704	-0.123	0.107	-0.080	0.094	
20	Myoglobin	Real	0.804	0.000	0.052	0.144						
		SOMCD	0.860	0.010	0.040	0.090	0.087	0.056	0.010	-0.012	-0.054	-0.029
		SELCON3-47	0.876	0.036	-0.018	0.142	0.060	0.072	0.036	-0.070	-0.002	
		SSNN-47	0.680	0.023	0.070	0.227	0.116	-0.124	0.023	0.018	0.083	
21	Papain	Real	0.260	0.169	0.175	0.396						
		SOMCD	0.190	0.350	0.110	0.340	0.552	-0.070	0.181	-0.065	-0.056	0.320
		SELCON3-47	0.200	0.267	0.229	0.343	0.481	-0.060	0.098	0.054	-0.053	
		SSNN-47	0.153	0.235	0.167	0.444	0.232	-0.107	0.066	-0.008	0.048	
22	Phosphoglycerate Kinase	Real	0.345	0.110	0.231	0.313						
		SOMCD	0.590	0.130	0.060	0.220	0.356	0.245	0.020	-0.171	-0.093	0.301
		SELCON3-47	0.589	0.085	0.107	0.244	0.281	0.244	-0.025	-0.124	-0.069	
		SSNN-47	0.714	0.022	0.078	0.185	0.055	-0.046	0.022	-0.027	0.049	
23	Pepsinogen	Real	0.205	0.386	0.165	0.243						
		SOMCD	0.160	0.420	0.100	0.310	0.270	-0.045	0.034	-0.065	0.067	0.044
		SELCON3-47	0.070	0.379	0.223	0.316	0.266	0.135	-0.007	0.058	0.073	
		SSNN-47	0.086	0.399	0.228	0.286	0.227	-0.119	0.013	0.063	0.043	
24	Prealbumin	Real	0.062	0.449	0.165	0.323						
		SOMCD	0.210	0.390	0.090	0.320	0.384	0.148	-0.059	-0.075	-0.003	0.138
		SELCON3-47	0.093	0.394	0.214	0.282	0.149	0.031	-0.055	0.049	-0.041	
		SSNN-47	0.152	0.427	0.192	0.229	0.245	0.090	-0.022	0.027	-0.094	
25	Rhodanase	Real	0.297	0.109	0.235	0.359						
		SOMCD	0.400	0.160	0.150	0.290	0.467	0.103	0.051	-0.085	-0.069	0.208
		SELCON3-47	0.368	0.168	0.185	0.284	0.323	0.071	0.059	-0.050	-0.075	
		SSNN-47	0.345	0.157	0.198	0.301	0.259	0.048	0.048	-0.037	-0.058	
26	Ribonuclease A	Real	0.210	0.331	0.218	0.242						
		SOMCD	0.270	0.370	0.090	0.270	0.357	0.060	0.039	-0.128	0.028	0.028
		SELCON3-47	0.223	0.256	0.220	0.306	0.578	0.013	-0.075	0.002	0.064	
		SSNN-47	0.184	0.290	0.228	0.297	0.329	-0.026	-0.041	0.010	0.055	
27	Substilin BPN	Real	0.302	0.178	0.225	0.295						
		SOMCD	0.380	0.260	0.100	0.260	0.408	0.078	0.082	-0.125	-0.035	0.305
		SELCON3-47	0.190	0.241	0.223	0.325	0.489	-0.112	0.063	-0.002	0.030	
		SSNN-47	0.325	0.165	0.207	0.302	0.103	0.023	-0.013	-0.018	0.007	
28	Substilin novo	Real	0.215	0.138	0.295	0.353						
		SOMCD	0.390	0.190	0.110	0.310	0.580	0.175	0.052	-0.185	-0.043	0.127
		SELCON3-47	0.341	0.200	0.199	0.275	0.659	0.126	0.062	-0.096	-0.078	
		SSNN-47	0.390	0.143	0.165	0.302	0.453	0.175	0.005	-0.130	-0.051	
29	Superoxide Dismutase	Real	0.018	0.367	0.298	0.316						
		SOMCD	0.210	0.310	0.110	0.370	0.648	0.192	-0.057	-0.188	0.054	0.145
		SELCON3-47	0.151	0.346	0.210	0.280	0.423	0.133	-0.021	-0.088	-0.036	
		SSNN-47	0.164	0.276	0.183	0.378	0.504	0.146	-0.091	-0.115	0.062	
30	T4 Lysozyme	Real	0.665	0.086	0.116	0.134						
		SOMCD	0.690	0.060	0.050	0.190	0.123	0.025	-0.026	-0.066	0.056	0.042
		SELCON3-47	0.660	0.061	0.092	0.175	0.045	-0.005	-0.025	-0.024	0.041	
		SSNN-47	0.642	0.050	0.091	0.217	0.082	-0.023	-0.036	-0.025	0.083	
31	Thermolysin	Real	0.415	0.165	0.215	0.206						
		SOMCD	0.470	0.160	0.070	0.300	0.297	0.055	-0.005	-0.145	0.094	0.067
		SELCON3-47	0.414	0.172	0.148	0.270	0.175	-0.001	0.007	-0.067	0.064	
		SSNN-47	0.478	0.093	0.114	0.315	0.230	0.063	-0.072	-0.101	0.109	
32	Tumor	Real	0.019	0.433	0.219	0.329						
	Necrosis	SOMCD	0.100	0.480	0.100	0.320	0.300	0.081	0.047	-0.119	-0.009	0.198
	Factor	SELCON3-47	0.197	0.542	0.091	0.063	0.377	0.178	0.109	-0.128	-0.266	
		SSNN-47	0.072	0.452	0.189	0.287	0.101	0.053	0.019	-0.030	-0.042	
33	Triose	Real	0.446	0.154	0.124	0.276						
	Phosphate	SOMCD	0.620	0.130	0.060	0.190	0.244	0.174	-0.024	-0.064	-0.086	0.128
	Isomerase	SELCON3-47	0.520	0.109	0.150	0.230	0.123	0.074	-0.045	0.026	-0.046	
		SSNN-47	0.518	0.112	0.136	0.234	0.116	0.072	-0.042	0.012	-0.042	
mean		Real										
		SOMCD					0.370					
		SELCON3-47					0.350					
		SSNN-47					0.381					
standard deviation		Real										
		SOMCD					0.162					
		SELCON3-47					0.273					
		SSNN-47					0.370					

Figures SI7 to SI54 show the 4 output plots of SSNN for the 48 proteins in the CDDATA.48 performed in LOOCV mode using the other 47 members of the reference set as the training set in each case. The parameters for all runs were: 28,000 iterations, a map size of 40×40, initial neighbourhood of 20, BMUs = 5, $L_0 = 0.1$, $t_1 = 7{,}000$ iterations, and a $k_1 = 5\times10^{-6}$.

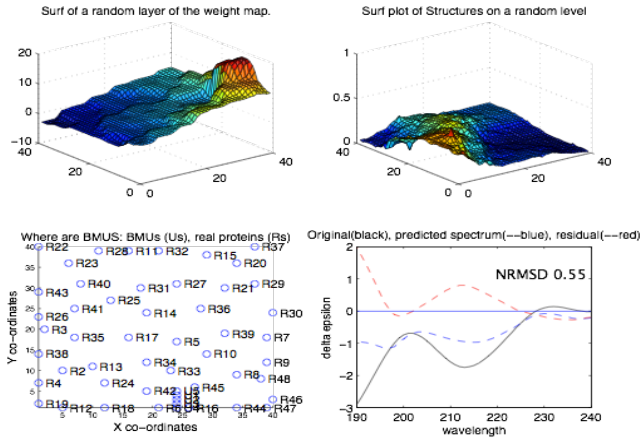


Figure SI7. Protein 1, α-Bungarotoxin SSNN-47 run as described in the text.

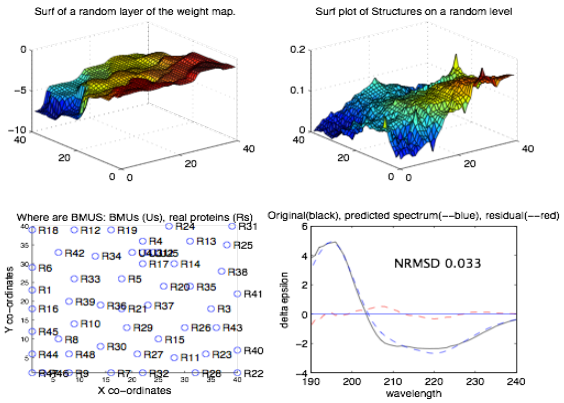


Figure SI8. Protein 2, Alcohol Dehydrogenase SSNN-47 run as described in the text.

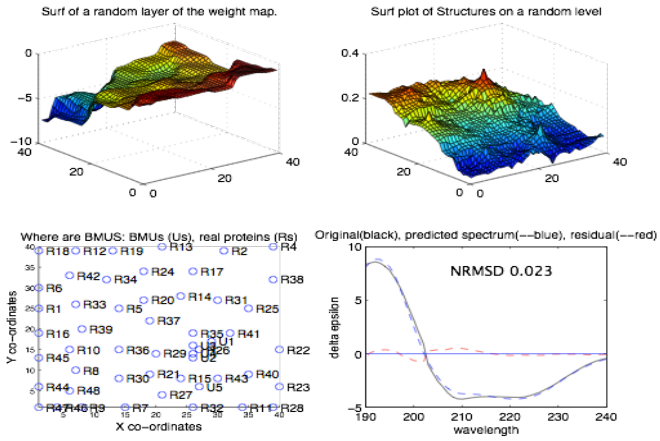


Figure SI9. Protein 3, Adenylate Kinase**.** SSNN-47 run as described in the text.

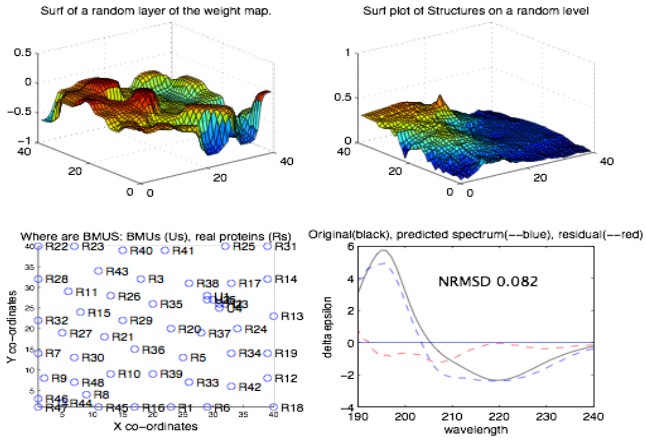


Figure SI10. Protein 4, Azurin. SSNN-47 run as described in the text.

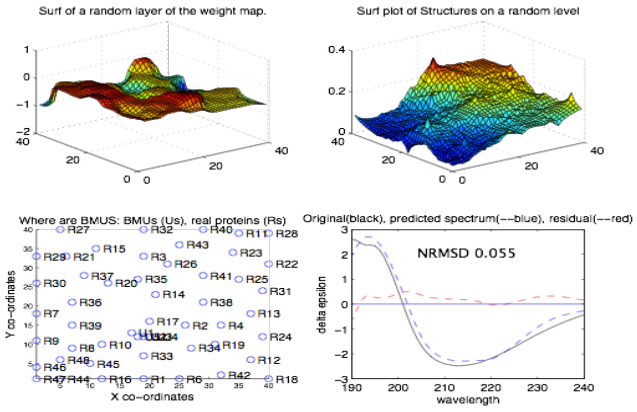


Figure SI11. Protein 5, β-lactoglobulin. SSNN-47 run as described in the text.

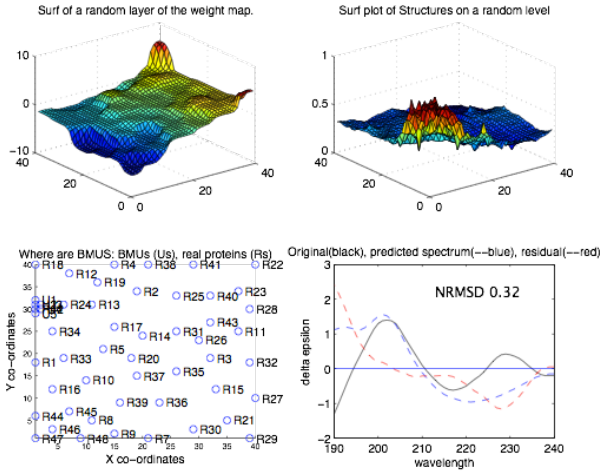


Figure SI12. Protein 6, Bence Jones Protein. SSNN-47 run as described in the text.

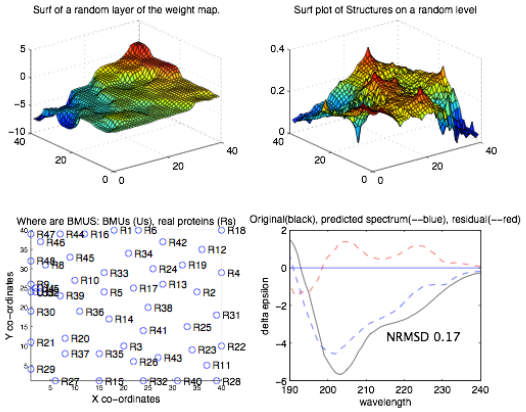


Figure SI13. Protein 7, Bovine Pancreatic Trypsin Inhibitor. SSNN-47 run as described in the text.

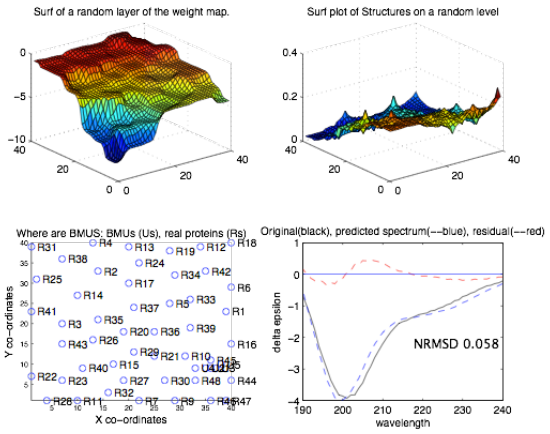


Figure SI14. Protein 8, Carbonic Anhydrase**.** SSNN-47 run as described in the text.

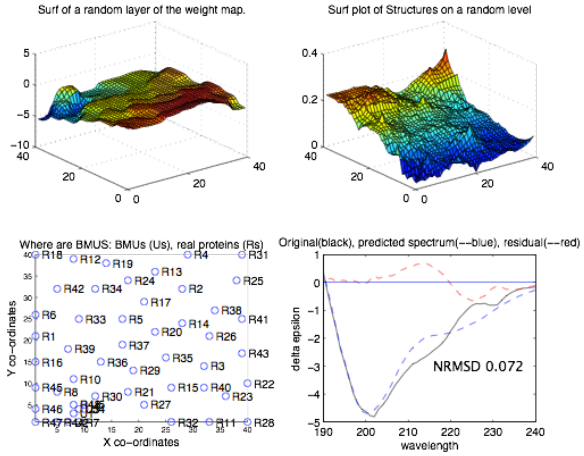


Figure SI15. Protein 9, CGA. SSNN-47 run as described in the text.

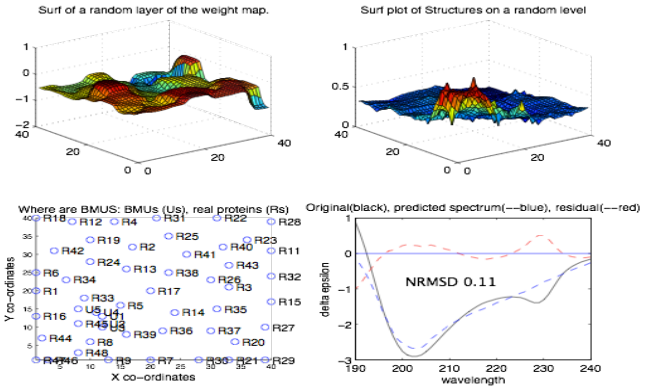


Figure SI16. Protein 10, α-Chymotrypsin. SSNN-47 run as described in the text.

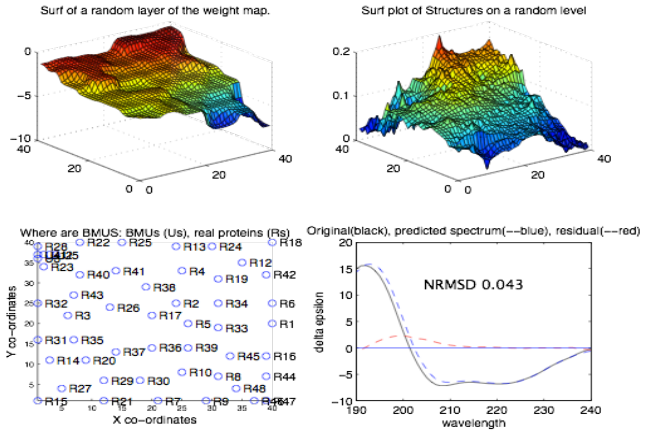


Figure SI17. Protein 11, Colicin A. SSNN-47 run as described in the text.

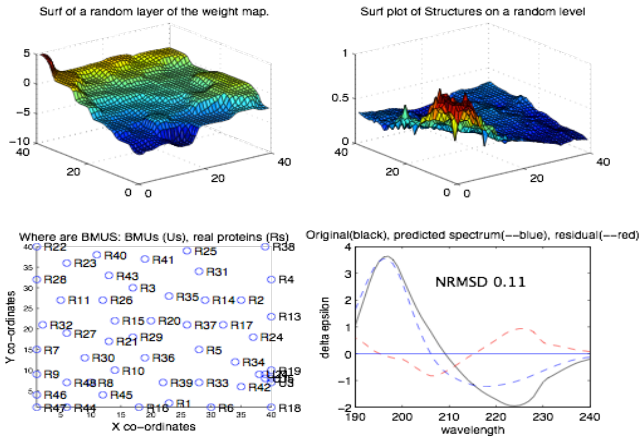


Figure SI18. Protein 12, Concanavalin A. SSNN-47 run as described in the text.

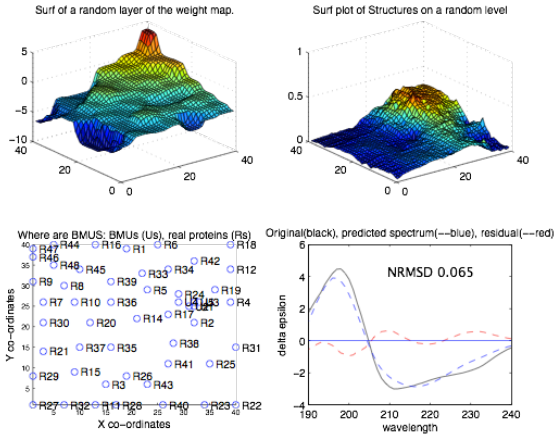


Figure SI19. Protein 13, Carboxypepsidase A. SSNN-47 run as described in the text.

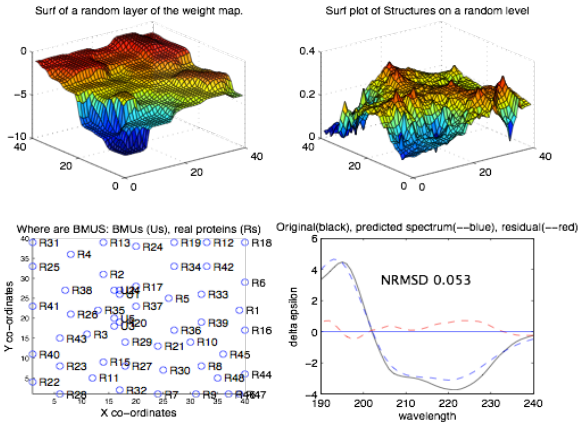


Figure SI20. Protein 14, Cytochrome C. SSNN-47 run as described in the text.

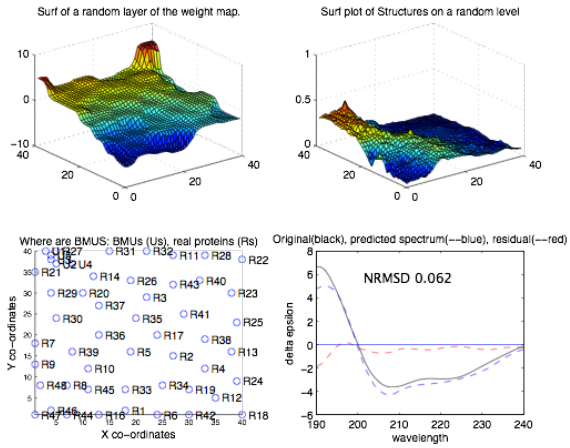


Figure SI21. Protein 15, EcoR1 Endonuclease**.** SSNN-47 run as described in the text.

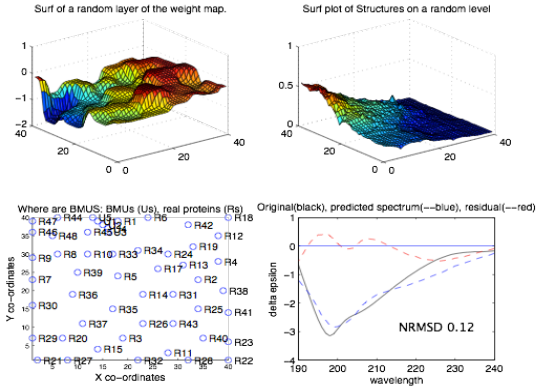


Figure SI22. Protein 16, Elastase. SSNN-47 run as described in the text.

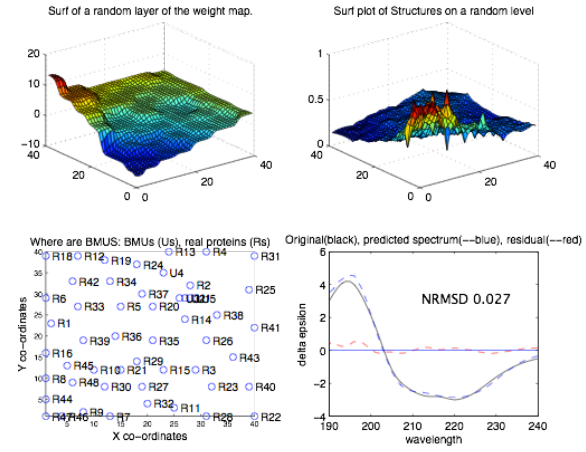


Figure SI23. Protein 17, Flavodoxin. SSNN-47 run as described in the text.

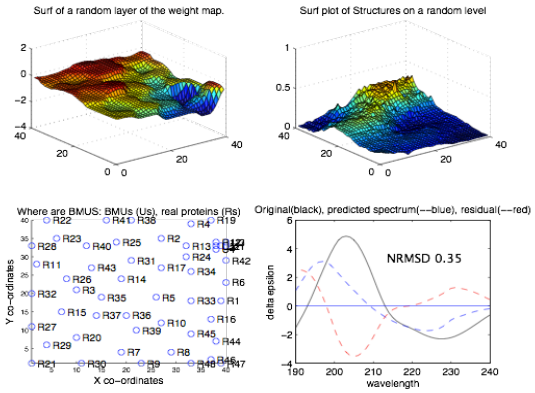


Figure SI24. Protein 18, γ-Crystalin. SSNN-47 run as described in the text.

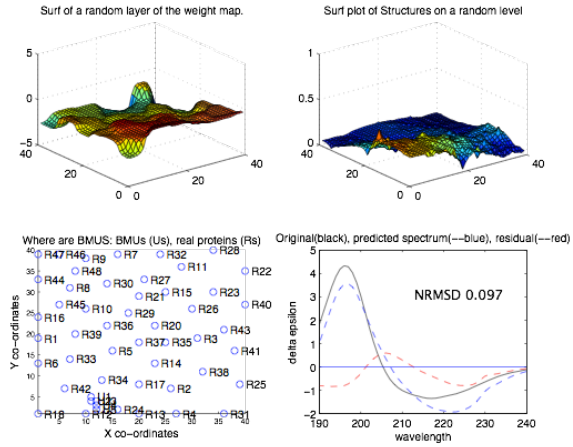


Figure SI25. Protein 19, Green Fluorescent Protein. SSNN-47 run as described in the text.

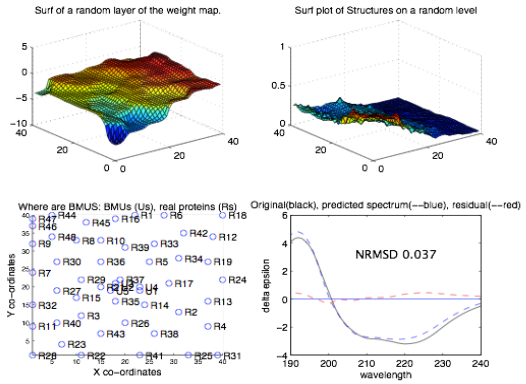


Figure SI26. Protein 20, Glyceraldehyde-3-phosphate dehydrogenase**.** SSNN-47 run as described in the text.

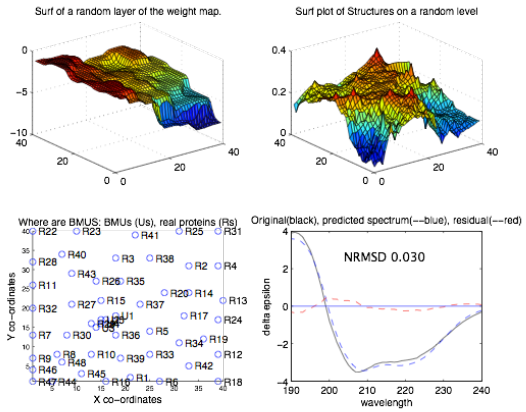


Figure SI27. Protein 21, Glutathione Reductase. SSNN-47 run as described in the text.

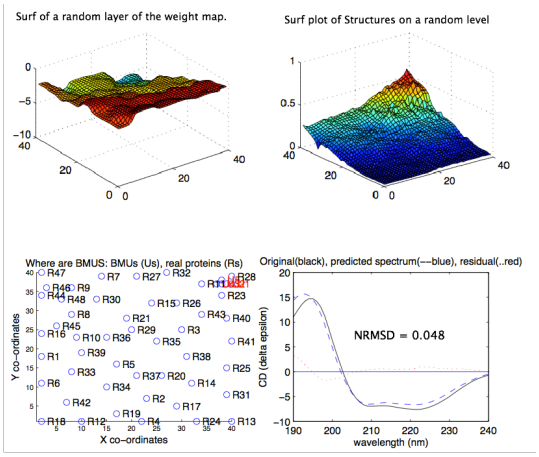


Figure SI28. Protein 22, Hemoglobin SSNN-47. run as described in the text.

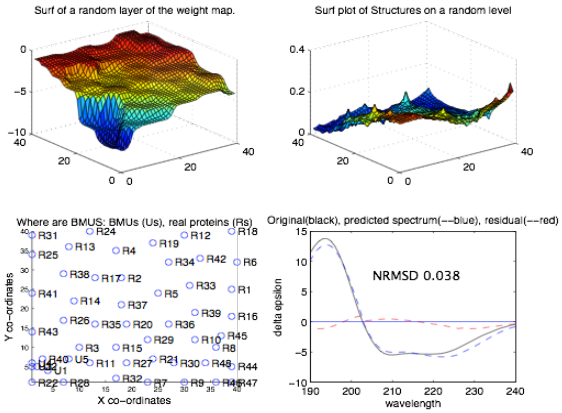


Figure SI29. Protein 23, Hemerythrin. SSNN-47 run as described in the text.

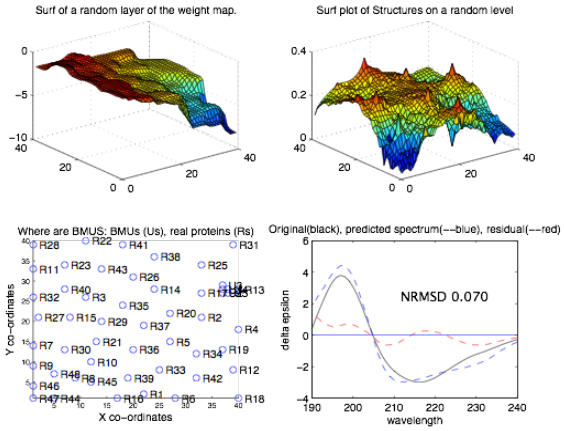


Figure SI30. Protein 24, Rat Intestinal Fatty Acid Binding Protein. SSNN-47 run as described in the text.

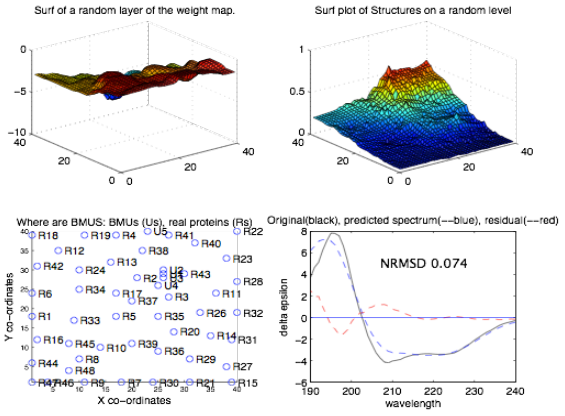


Figure SI31. Protein 25, Insulin. SSNN-47 run as described in the text.

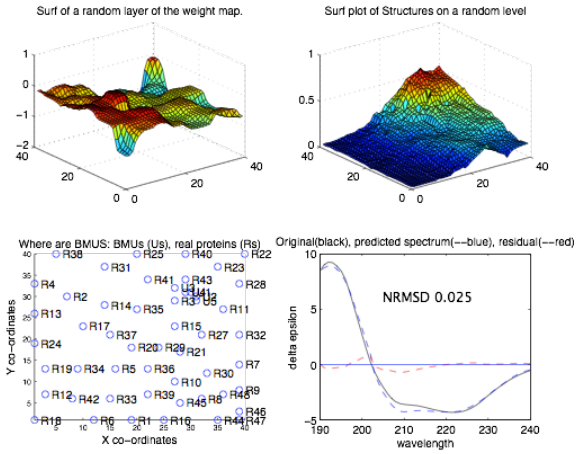


Figure SI32. Protein 26, Lactate Dehydrogenase SSNN-47 run as described in the text.

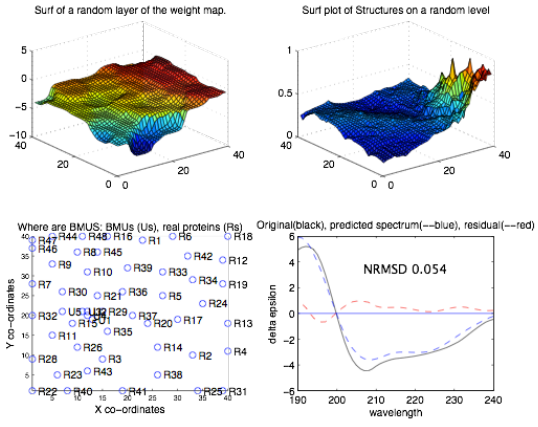


Figure SI33. Protein 27, Lysozyme SSNN-47 run as described in the text.

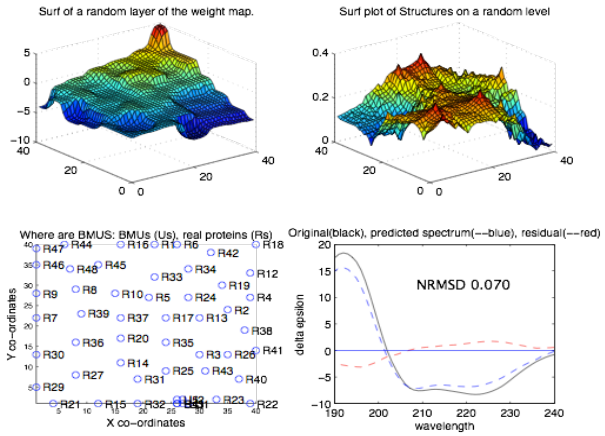


Figure SI34. Protein 28, Myoglobin SSNN-47 run as described in the text.

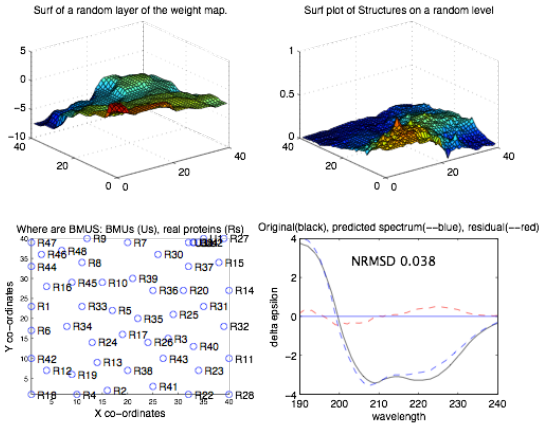


Figure SI35. Protein 29, Nuclease SSNN-47 run as described in the text.

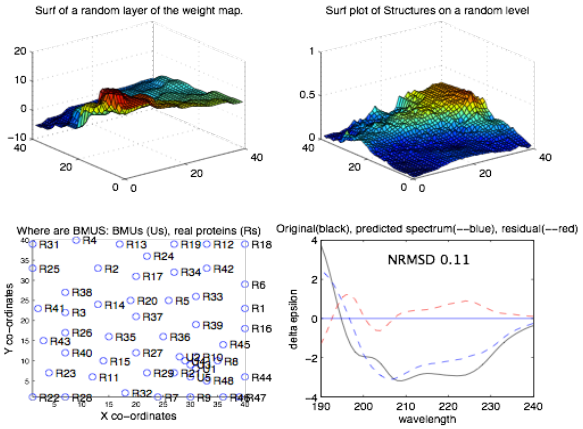


Figure SI36. Protein 30, Papain SSNN-47 run as described in the text.

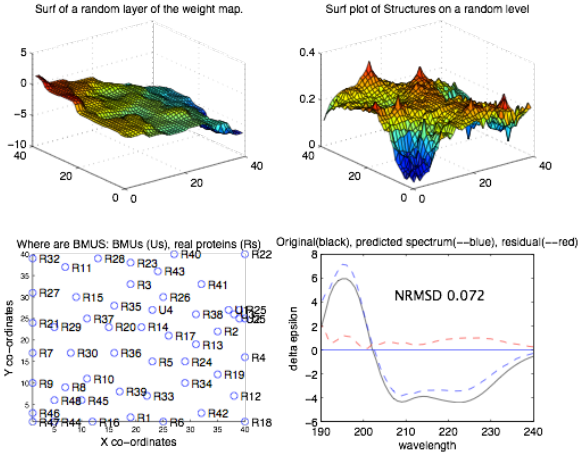


Figure SI37. Protein 31, Parvalbumin SSNN-47 run as described in the text.

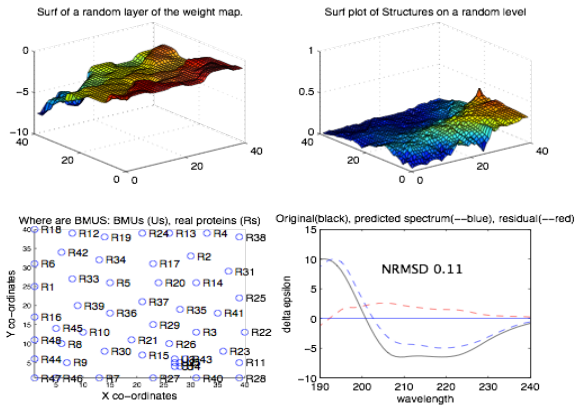


Figure SI38. Protein 32, Phosphoglycerate Kinase SSNN-47 run as described in the text.

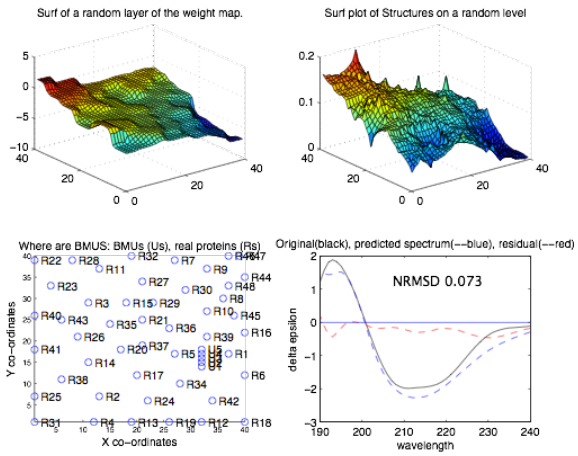


Figure SI39. Protein 33, Pepsinogen SSNN-47 run as described in the text.

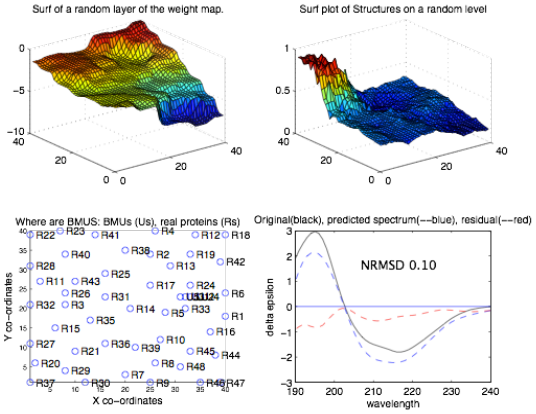


Figure SI40. Protein 34, Prealbumin SSNN-47 run as described in the text.

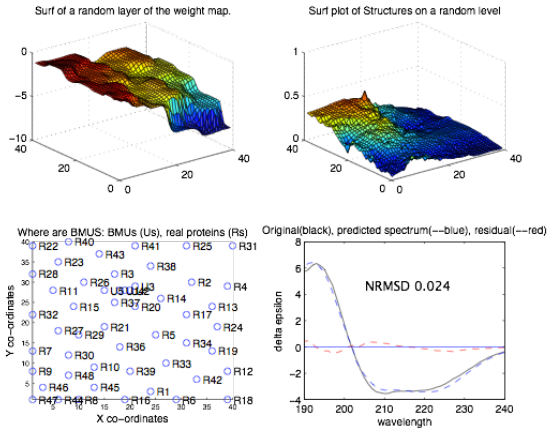


Figure SI41. Protein 35, Rhodanase SSNN-47 run as described in the text.

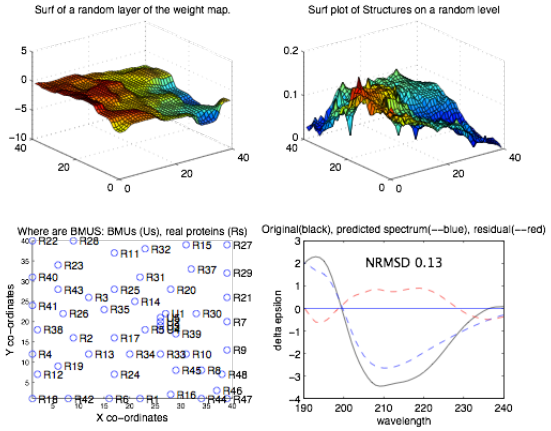


Figure SI42. Protein 36, Ribonuclease A SSNN-47 run as described in the text.

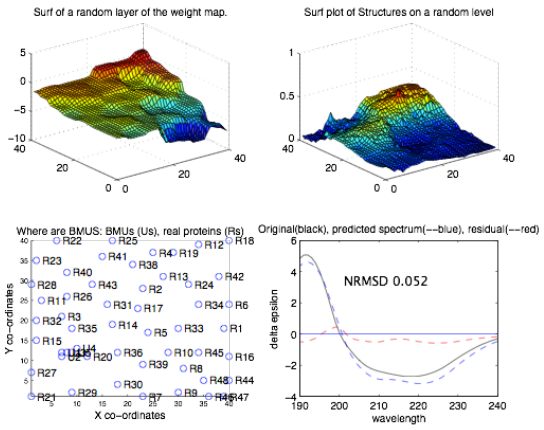


Figure SI43. Protein 37, Substilin BPN SSNN-47 run as described in the text.

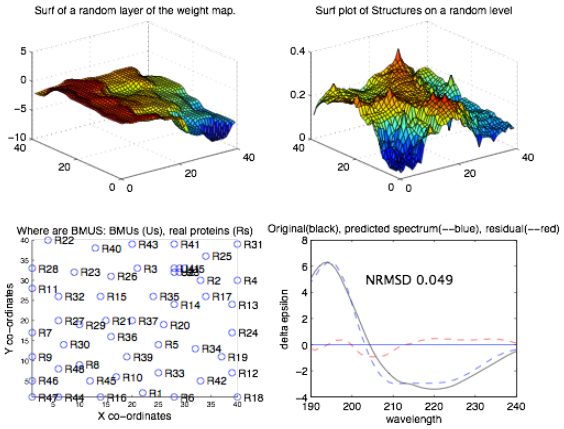


Figure SI44. Protein 38, Substilin novo SSNN-47 run as described in the text.

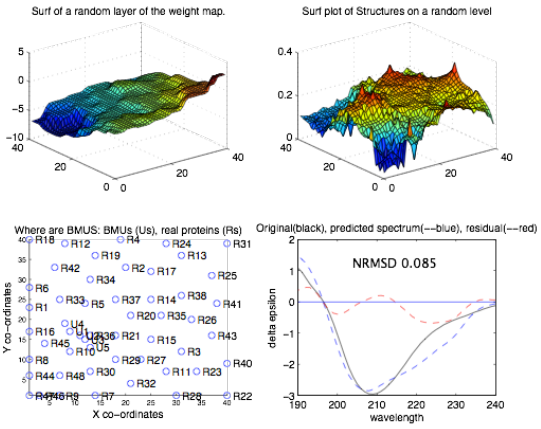


Figure SI45. Protein 39, Superoxide Dismutase SSNN-47 run as described in the text.

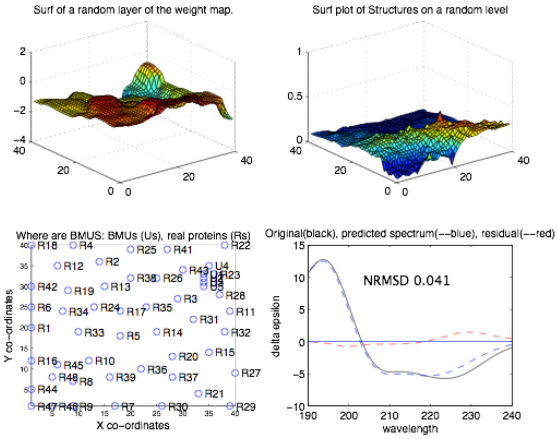


Figure SI46. Protein 40, T4 Lysozyme SSNN-47 run as described in the text.

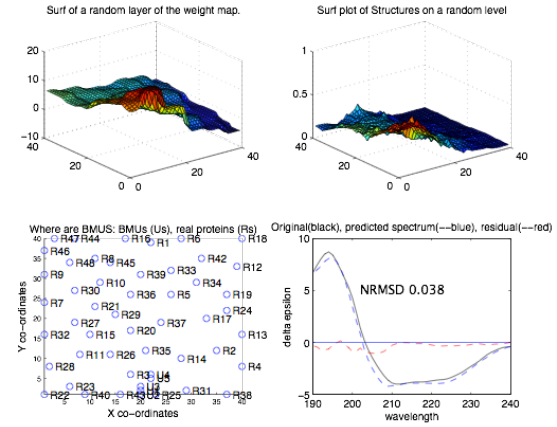


Figure SI47. Protein 41, Thermolysin SSNN-47 run as described in thetext.

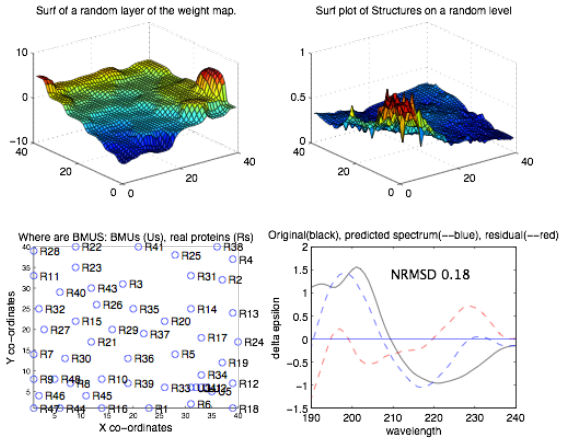


Figure SI48. Protein 42, Tumor Necrosis Factor SSNN-47 run as described in the text.

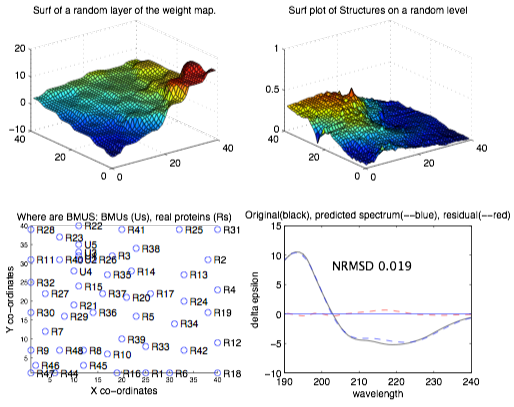


Figure SI49. Protein 43, Triose Phosphate Isomerase SSNN-47 run as described in the text.

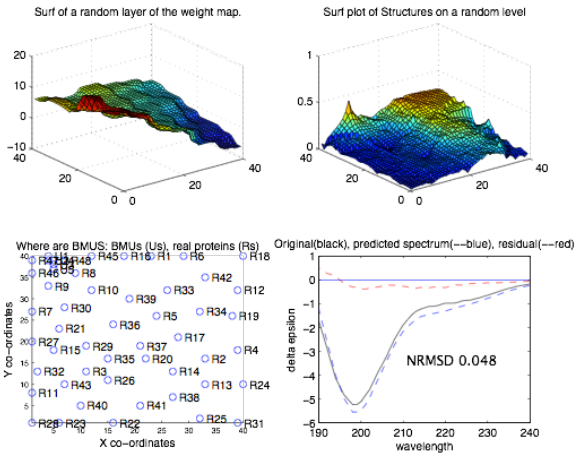


Figure SI50. Protein 44, Apo-cytochrome C (5°C) denatured SSNN-47 run as described in the text.

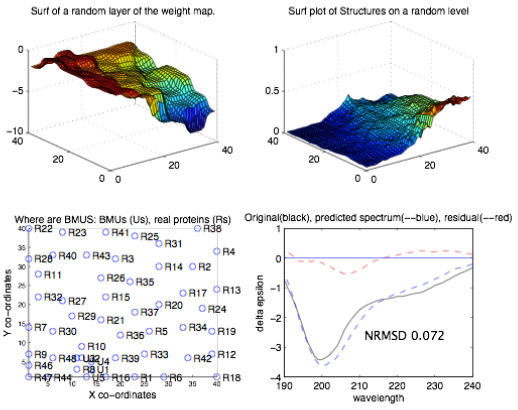


Figure SI51. Protein 45, Apo-cytochrome C (90°C) denatured e SSNN-47 run as described in the text.

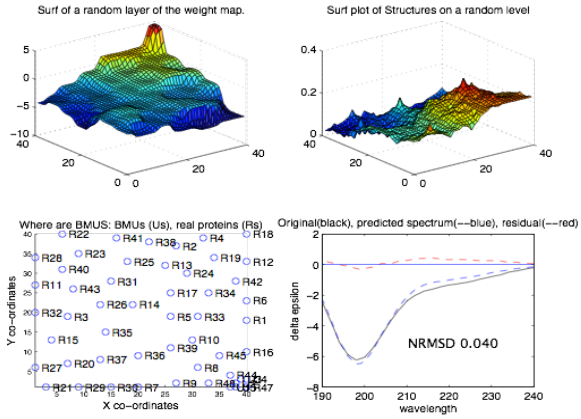


Figure SI52. Protein 46, Ribonuclease (20°C) denatured SSNN-47 run as described in the text.

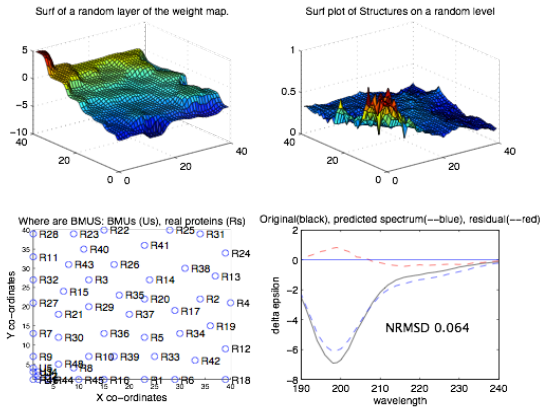


Figure SI53. Protein 47, Staphyllococcal Nuclease (6°C) denatured SSNN-47 run as described in the text.

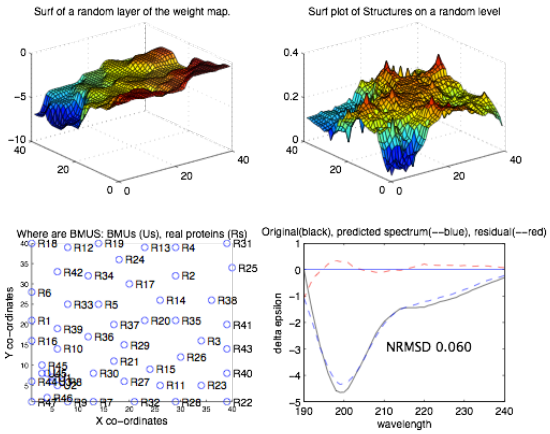


Figure SI54. Protein 48, Staphyllococcal Nuclease (70°C) denatured SSNN-47 run as described in the text.

1. Whitmore L, Wallace BA. DICHROWEB: an online server for protein secondary structure analyses from circular dichroism spectroscopic data. Nuc. Acids Res. 2004; 32:W668-673.

Protein Secondary Structure Prediction from Circular Dichroism Spectra Using a Self-organising Map with Concentration Correction

VINCENT HALL,[1] MEROPI SKLEPARI,[2] AND ALISON RODGER[2*]

1. MOAC, Department of Chemistry and School of Engineering, University of Warwick, Coventry CV4 7AL, UK.
2. Warwick Centre for Analytical Science and Department of Chemistry, University of Warwick, Coventry, CV4 7AL, UK. Phone: +44 2476574696. Fax: +44 2476575795. Email: A.Rodger@warwick.ac.uk

Short title: Secondary structure from CD using a SOM

KEY WORDS: Artificial Neural Network, Kohonen map, "Secondary Structure Neural Network", SSNN, peptides, MATLAB, CDPro, Dichroweb

Contract grant sponsor: EPSRC; Contract grant number: EP/F500378/1.

ABSTRACT Collecting circular dichroism (CD) spectra for protein solutions is a simple experiment, yet reliable extraction of secondary structure content is dependent on knowledge of the concentration of the protein—which is not always available with accuracy. We previously developed a self-organising map (SOM), called Secondary Structure Neural Network (SSNN), to cluster a database of CD spectra and use that map to assign the secondary structure content of new proteins from CD spectra. The performance of SSNN is at least as good as other available protein CD structure fitting algorithms. In this work we apply SSNN to a collection of spectra of experimental samples where there was suspicion that the nominal protein concentration was incorrect. We show that by plotting the normalized root mean square deviation of the SSNN predicted spectrum from the experimental one versus a concentration scaling-factor it is possible to improve the estimate of the protein concentration while providing an estimate of the secondary structure. For our implementation (51 data points 240 – 190 nm in nm increments) good fits and structure estimates are obtained if the NRMSD (normalised root mean square displacement, RMSE/data range) is < 0.03; reasonable for NRMSD < 0.05; and variable above this. We have also augmented the reference database with 100% helical spectra and truly random coil spectra.

INTRODUCTION

To extract secondary structure information for globular proteins from circular dichroism (CD) spectra, expert opinion must be sought; this is usually from either a person who has worked in the field for a long time, or a software methodology. A number of such methodologies are available in Dichroweb [1,2] and at the CDPro website.[3,4] However, all such available methodologies are dependent on the accuracy of the protein concentration. In other papers [5,6] we reported the development of SSNN, "Secondary Structure Neural Network", which is a software package to assign secondary structures using a self-organising map (SOM) methodology. Our approach is similar in intent to the family of K2d programs [7] but more flexible in terms of reference data set and wavelength range. It has also been validated by testing it in a leave-one-out methodology using the CDDATA.48 reference set from CDPro [3,4] as a 47-member training set with one test protein, repeating the test 48 times and comparing with CDSSTR, SELCON3, and K2d.[5,6] CDDATA.48 has structure vectors associated with it and may be found on the CDPro website (http://lamar.colostate.edu/~sreeram/CDPro/main.html), which is maintained by Sreerama *et al*. at Colorado State University.[8] The structure labels used are written as a vector

throughout this work and refer to: (α-helix, distorted α-helix, β-sheet, distorted β-sheet, turn, other) as in references.[8,9,10] The assignments come from DSSP annotation with the 2 residues at each end of helices and 1 residue at each end of β-strands being taken as distorted.[11] In this work the structure vectors are quoted as fractions of 1 in this order. Total α-helix content is thus the sum of the first two components and total β-sheet content is the sum of the third and fourth components. Pure CD structure types produce spectra similar to those seen in Figure 1.

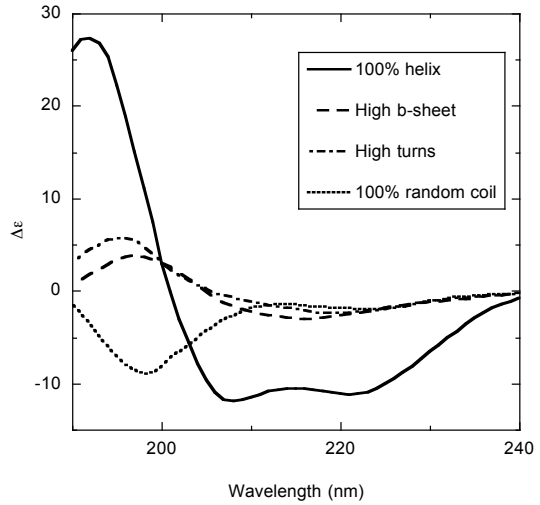


Figure 1: The CD spectra of the proteins of reference set CDDATA.(48+5) with the most extreme structure types: an extrapolation of the CD of peptide Aurein 2.5 to model 100% α-helix; rat intestinal fatty acid binding protein is the highest β-sheet content in the reference set (58.4%); Azurin has the largest turn content in the reference set (31.2 %); N-formyl acetic acid is 100% random coil protein.

SSNN proceeds in three distinct units. In the first unit, SSNN1, the spectra of a reference set of known proteins (with known structures) is organised on a map of chosen size so that similar spectral shapes are neighbours. All nodes on the map are given spectra interpolated between those of the original reference set. In SSNN2, secondary structure vectors corresponding to the spectrum of that node are assigned to all nodes. In SSNN3, the best position on the map for the spectrum of an unknown protein is found and its structure vector determined from its position. For a given reference set, SSNN1 and SSNN2 need only be run once.

SSNN performed at least as well as other available fitting methodologies in our earlier work.[5,6] Its worst structure suggestions were where the unknown protein was on the edge of the structures map or where the intensity in the 208–222 nm region gave a relatively small spectral NMRSD (normalised root mean square displacement, see below) but the shape of the experimental spectrum was reminiscent of an α-helix and that of the predicted model spectrum β-sheet (or conversely). Our motivation in developing a new structure fitting method was to have an approach that could be used in a wide variety of situations. Our first attempts to apply SSNN 'in the real world' were not entirely successful for two completely different reasons. The first reason was that some proteins and peptides (particularly the latter) ended up on the very edge of the spectral map because our reference set did not include very high helical or completely unfolded spectra. The second was that despite our best efforts and those of our colleagues, estimates of protein concentration were never as accurate as we thought. This has the automatic consequence for any fitting methodology of introducing an error into the estimates of secondary structure.

In this work we have therefore augmented the SSNN reference set with 5 spectra created to represent 100% α-helical proteins and 100% 'random coil', bringing the reference set up to 53 proteins. Here random coil refers to the spectrum observed for an unfolded peptide. Its structure vector is 100% 'other', *i.e.* (0,0,0,0,0,1). We have also developed a way of using SSNN to improve the estimates of protein concentration.

METHODS

A SOM is a type of unsupervised neural network that takes high-dimensional data (in our case CD spectra) and clusters them, then visualises it in a manner that is much easier to understand than the high-dimensional data set. SSNN is described in reference [5] and the details of how to implement the code are given in [6]. The process of training the SOM with a reference data set (SSNN1) is first to make a matrix of $n \times n$ (in this case 40×40) vectors containing pseudo-random numbers. Then one selects, at random, a protein spectrum from the reference set and compares it with all of the random spectra in the matrix. The random vector in the 'spectra map' that has the smallest Euclidean distance from the selected protein spectrum is called the best matching unit (BMU) and is the 'winner'. Next this BMU is made more similar to the selected spectrum using a learning rule, which makes the numbers in the random vector more similar to the protein spectrum. At the same time the vectors in the map near the BMU, the neighbourhood, are made more similar to the protein spectrum as well, but to a lesser degree in a distance (map coordinate)-dependent way. This is done for thousands of iterations, 28000 in our case. Once this is finished, the spectra map is trained, and ready for the next stage. Due to the random selection, the physical appearance of the maps change each time SSNN1 is run, though the clustering will be the same and thus ultimate fits should not change. One layer of a trained SOM can be seen in Figure 2a for our augmented version of CDDATA.48 which we denote CDDATA.(48+5). The figure shows the clustered CD spectra in a 40×40 spectra map for the 222 nm data point. This only shows one of the stack of 51 data points for each spectrum, so there are 50 other nm points for each spectrum, and thus a stack of 50 other maps could be produced. The map shows the 53 reference set spectra, and also virtual interpolated spectra filling in the gaps to make up the 1600 spectra.

The second module, SSNN2 takes the clustered spectra and constructs a structures map by finding the coordinates of the BMUs of the 53 proteins in the spectra map and placing their structures at the same coordinates in the structures map. For the virtual structures, SSNN2 takes a distance-weighted sum of 5 of the structures of neighbouring spectra from the reference set. A typical result is shown in Figure 2b for the α-helix. There is a structures map for each of the 6 structure types used in this work. The two peaks in Figure 2b in this map show that not all α-helix-rich proteins have the same spectra and hence structure vectors.

In SSNN3, a model of a test spectrum is determined (see Results for examples) as a weighted sum of its 5 BMUs (positions of BMUs on the SOM are also given in the output files). The structures then follow from the same weighted sum as in the structures map. The model or predicted (fitted) spectrum has an NRMSD (normalised root mean squared deviation) associated with it, and this is used to indicate how much the structures prediction can be trusted. We use these definitions of RMSD and NRMSD,

$$RMSD = \sqrt{\frac{\sum_{i=1}^{N}\left(S_i - M_i\right)^2}{N}} \qquad (1)$$

$$NRMSD = \frac{RMSD}{M_{\max} - M_{\min}} \qquad (2)$$

where S_i are the elements of the real spectrum, and M_i are elements of the model spectrum, N is the number of data points in a spectrum (51 in this case). M_{max} and M_{min} are the largest and smallest observed values, in this case the largest and smallest values in the model spectrum being evaluated.

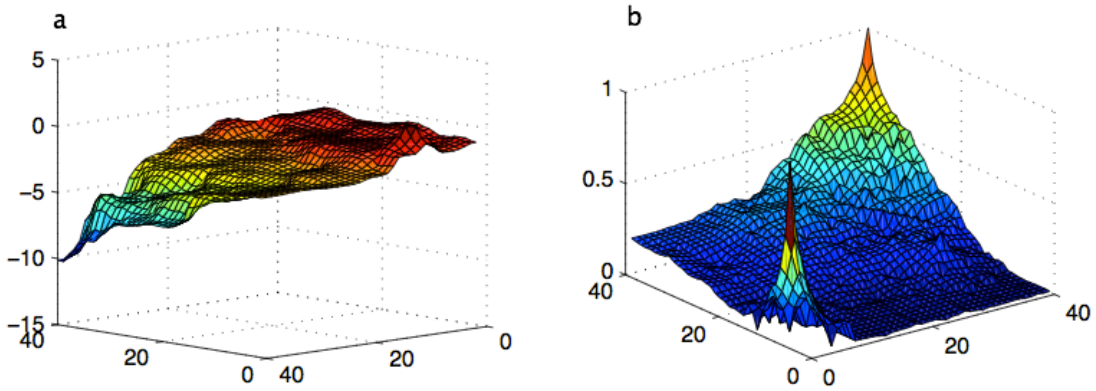


Figure 2: (a) Spectral intensity map for 222 nm CD signals. (b) Structures map for α-helix structure vector component after optimisation starting from a reference set of 53 proteins which includes CDDATA.48 from CDPro and an additional 5 spectra (see text).

In this implementation of SSNN we have chosen to represent CD spectra as vectors of 57 numbers, the first 51 being the $\Delta\varepsilon$ intensities (where the concentration is that of amino acid residues not protein molecules) for 240–190 nm in 1 nm steps and the last 6 being the structure vector components. In our previous work [5] we showed that with a reference set of 48 spectra, the SOM size of 40×40 nodes or spectra was optimal. The same will be true for 53 spectra, though a large increase or decrease would require a larger or smaller map. A version of SSNN (SSNNGUI) is available pre-trained with CDDATA.(48+5) at http://www2.warwick.ac.uk/fac/sci/chemistry/research/arodger/arodgergroup/research_intro/instrumentation/ssnn/ and the instructions for its use are given in details in reference [6]. The test spectra must then be matching 51-number column vectors. Alternatively SSNN1_2.app may be trained on reference sets for any wavelength range or set of proteins, as also described in reference [6]. The key innovation of this work is to have made both SSNNGUI and SSNN1_2.app useable when one only has an estimate of the protein concentration. In this case a concentration scaling factor range should be entered on the GUI (graphical user interface) and also a step size. Bearing in mind that more calculations take longer to perform, it is preferable to do a coarse-grained calculations first then refine the step size for a smaller range. A spectral NRMSD against concentration scaling factor plot will be additional output if these parameters are entered on the GUIs.

Insulin and polylysine sample preparation and data collection

Insulin and polylysine were obtained from Sigma–Aldrich (insulin from bovine pancreas I6634, polylysine P 4707 MW 70000–150000). For the pH 2.3 insulin, 0.44 mg of insulin was dissolved in NaOH (0.1 M). Sodium phosphate buffer (to final concentration 4 mM) was added, resulting in an insulin solution of ~0.3 mg/ml (concentration calculated by measuring the UV absorbance at 278 nm and using the Sigma–Aldrich extinction coefficient of 6080 $mol^{-1}cm^{-1}dm^3$). The pH was adjusted to 2.3 with $HClO_4$ 0.1M and it was diluted by factor of 4. Polylysine was made to nominal 0.10 mg/mL in water and adjusted to pH 11.2 with NaOH resulting in a nominal 0.094 mg/mL solution.

For the UV absorbance measurements, a Jasco V-660 spectrophotometer was used. All the CD spectra were taken in a Jasco J-815 or J-715 CD spectropolarimeter. The balance used was a Mettler Toledo XP2U and the pH meter Mettler Toledo Seven Compact pH/Ion S220 InLab Nano Sensors.

RESULTS AND DISCUSSION

Applying SSNN3 to real data

Following our successful leave-one-out validation of SSNN using spectra from the CDPro website,[4] we embarked on applying SSNN to real data. This project had mixed success until we

considered the fact that although the proteins all had a nominal value of 0.1 mg/mL, this concentration was unlikely to be correct. We therefore wrote a version of SSNN3 that processes an input spectrum through SSNN several times, each time multiplying the spectrum vector by different factors, which we call the 'concentration scaling factors'. As illustrated in the examples below, we found that the plot of NRMSD versus scaling factor often has a single minimum. If the value of the minimum NRMSD is small (<0.03) we are confident this scaling factor gives a reasonable estimate of the true concentration and secondary structure estimate. It is advisable to view the output for a number of scaling factors near the minimum if the structure estimates differ significantly—this is particularly true if *e.g.* the low wavelength data quality is poor. For larger NRMSDs, visual inspection of the overlay of model and experimental spectra is advisable as discussed below. Better fits also correlate with the BMUs not jumping around the SOM.

For ease of use we have made a single GUI option for using SSNN3 pre-trained with reference set CDDATA.(48+5). We previously used it with one single concentration but have added the option to scale the concentration automatically. This application is denoted SSNNGUI.app. More advanced users can use a re-trainable version, which includes SSNN1 and SSNN2 as well as SSNN3, called SSNN1_2.app. SSNN is written in MATLAB, and for the GUI, MATLAB's GUIDE (graphical user interface development environment) was used. SSNN3-single, runs SSNN once for each unknown protein in the test set to determine the secondary structures of the proteins in question at the stated concentration. SSNN3-multiple allows the user to select a range of concentration scaling factors and gives secondary structure analyses and NRMSD values for each scaling factor each as part of the output. The GUI of SSNN3 takes less than 2 seconds to run (per spectrum) on a 2009 MacBook Pro laptop with 8 GB RAM, and a 2.26 GHz processor once the MATLAB Compiler Runtime is installed (for details see reference [6]). The training process of SSNN1 and SSNN2 running for 28,000 iterations takes about 20 minutes to run. SSNNGUI is available for Mac and Windows at http://www2.warwick.ac.uk/fac/sci/chemistry/research/arodger/arodgergroup/research_intro/instrumentation/ssnn/.

Guide lines for the formatting of the input files can be found in the instructions text file on the SSNN website. The output is plots to show where the BMUs making the model spectrum are relative to the reference set members, an overlay of the model (or predicted) spectrum and the original experimental spectrum, along with the spectral NRMSD value. When SSNNGUI is run in multiple mode, a plot of NRMSD versus scaling factor is also produced. The results reported in this paper have been determined using the SSNNGUI.

Applications

The remainder of this paper is structured around particular examples that illustrate aspects of the performance of SSNN. The first two examples are for highly helical and highly sheet proteins followed by the mixed structure insulin at low pH as a function of temperature to illustrate the strengths and weaknesses of applications of SSNN to solution-phase protein structure and concentration determination. Insulin was chosen as a worst-case scenario as discussed below. The examples are followed by illustrations with unknown lipoproteins, ZapA mutants (the protein that bundles FtsZ fibres), some related toxins, and concludes with a peptide in different solvents.

Membrane peptide: α-helix

The experimental CD, scaling factor 1.0 model spectrum, SOM and plot of spectral NRMSD vs scaling factor for an α-helical peptide whose concentration was known fairly accurately are shown in Figure 3. The suggests 0.95 (83% helix) and 1.0 (84% helix) are best fits, with 0.9 (81%) and 1.05 (85%) still having very good fits. All have 0% β-sheet content (Table 1). We therefore conclude this peptide has 83±2% helix, 0% sheet, 5±1% turn and 12±1% Other. As discussed below for mixed structure systems it may be appropriate to declare a bigger uncertainty.

Table 1: Secondary structure estimates for an α-helical peptide as a function of scaling factor.

Concentration scaling factor	α-Regular	α-Distorted	β-Regular	β-Distorted	Turn	Other
0.9	0.624	0.184	0.003	0.002	0.052	0.136
0.95	0.655	0.176	0	0	0.048	0.121
1	0.67312	0.16736	0	0	0.045	0.114
1.05	0.69919	0.15312	0	0	0.043	0.105

(a) (b)

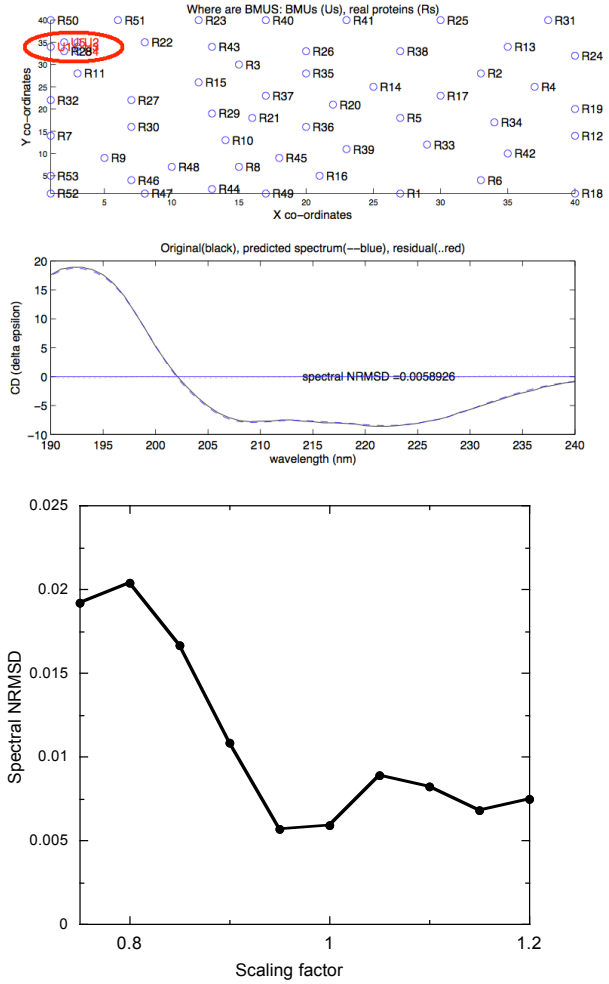


Figure 3: SSNN results for a highly helical peptide in a lipid environment. (a) The SOM with BMUs indicated clustered on the top left of the map (spectra numbers are in the order provided by CDDATA.48), the experimental spectrum (assuming 0.1 mg/mL concentration) and model spectrum for scaling factor 1.0. (b) SSNN spectral NRMSD.

Polylysine: β-sheet

Polylysine (~1.1 mg/mL) was dissolved in H_20 and the pH was adjusted to 11.4 with NaOH then heated at 55°C for 30 min to produce a β-sheet (Figure 4).[12,13] The spectral NRMSD (Figure 4b) is a minimum for scaling factor 0.9 where the structure vector is (0.020,0.052,0.29,0.14,0.20,0.29). With this highly sheet protein (and others we tested, data not shown), the accuracy of the concentration is not a great concern as the α-helical percentage was 7% and β-sheet percentage was 44% for concentration scaling factors ranging from 0.65–1.2. The local minima in the NRMSD plots at higher scaling factors (~1.85) are clearly bad fits when inspected as they look helical.

We chose to analyse polylysine because it is a simple system which is deemed to give the archetypical α-helix, β-sheet, and random coil spectra under different conditions. After extensive work (data not shown) we remain unconvinced that the pH 11.2 room temperature polylysine structure is a pure α-helix, at least in our laboratory.

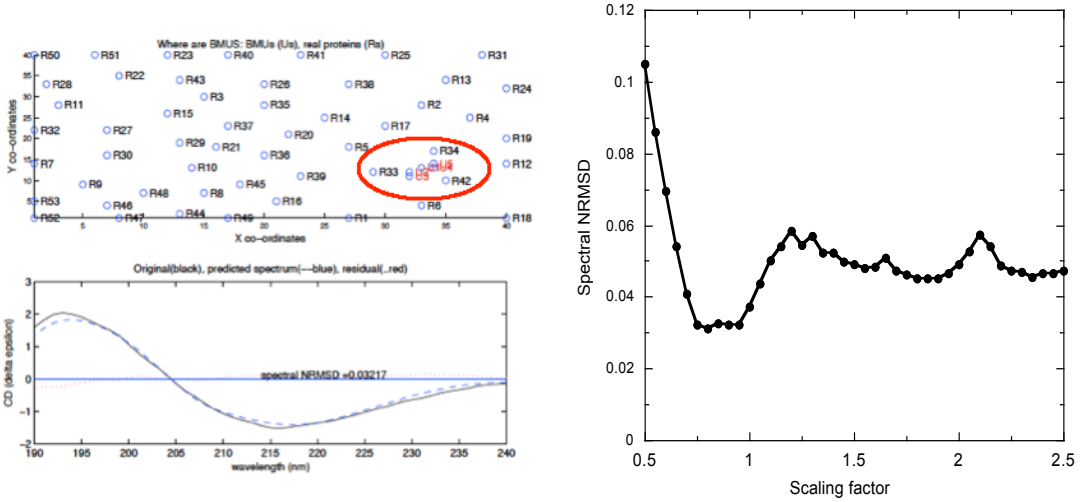


Figure 4: SSNN results for a highly sheet protein. (a) The SOM with BMUs indicated clustered towards the bottom right of the map (spectra numbers are in the order provided by CDDATA.48), the experimental spectrum (assuming 0.1 mg/mL concentration), and model spectrum for scaling factor 1.0. (b) SSNN spectral NRMSD.

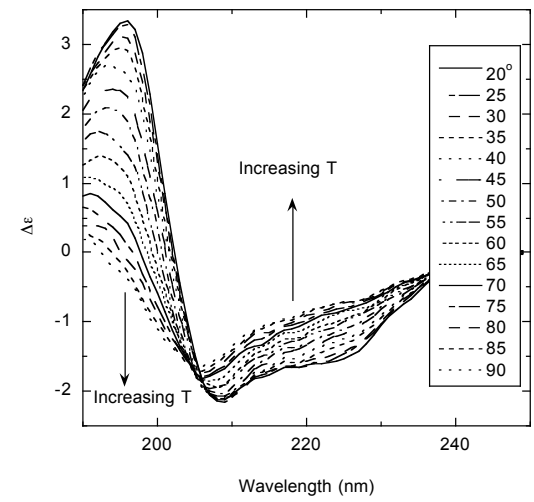
Insulin

Insulin is a small 2-peptide protein whose crystal structure for the neutral pH zinc containing protein (PDB entry 4INS) was annotated to have structure vector (0.29, 0.23, 0.02, 0.04, 0.05, 0.36) for Woody and Sreerama's data base.[3] Insulin is a challenging protein to get and keep in solution, and its structure varies with pH and whether it has zinc present or not. Despite being extensively studied, its structural details remain unclear in some environments, so resulting data will be useful. In addition, in leave-one-out testing with SSNN and SELCON3 this fairly helical protein was the third worst structure analysis.[5] The results shown below are for insulin at pH 2.4, which is a zinc-free structure as it illustrates the limitations of SSNN effectively. The spectral shape is sufficiently different from the neutral pH zinc containing structure so the data base not to be modified.

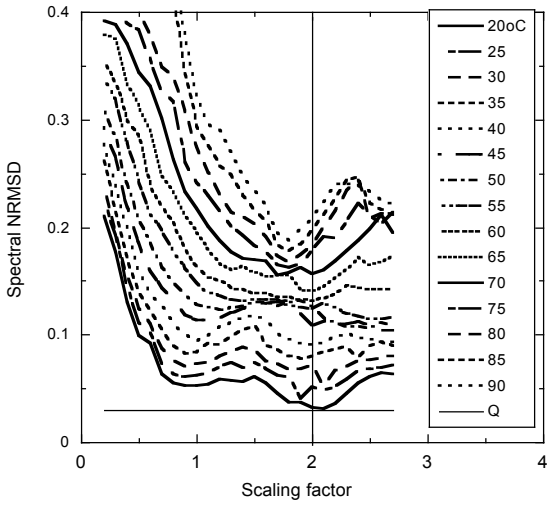
CD spectra for pH 2.4 insulin are shown in Figure 5a as a function of temperature (assuming nominal 0.1 mg/mL concentration). Our aim here was to have a constant concentration for a series of spectra. In practice, a small degree of evaporation did increase the concentration for the last few spectra. The SSNN spectral NRMSDs are plotted in Figure 4b on a displaced vertical scale and the predicted helical and sheet content are plotted in Figure 4c as a function of temperature using scaling factor 2.0 for the first 11 spectra and 1.8 for the last 4 (some evaporation occurred during the experiment as seem by the absorbance signals). The room temperature structure predictions are ~9% less helical than both the crystal and the zinc-free pH 7.3 solution data (not shown) which is in accord with the available low pH NMR data (PDB 2HIU,[14] for insulin with the B chain carboxy-terminus native alanine mutated to threonine) which is 39% helix (according to DSSP annotation).

The peculiar step in Figure 5c between 25°C and 30°C provides an interesting illustration of another value of the variable concentration methodology as a means of estimating reliability of structure predictions. As shown in Figure 5b, none of the spectral NRMSDs are below the somewhat arbitrary quality threshold of 0.03. The factor 2.0 BMUs and fits are illustrated in Figures 5d and e for 25°C and 30°C and the factor 2.1 for 30°C. The BMUs for the two 2.0 fits are in slightly different parts of the map and that for 2.1 spans both parts. This instability reflects the fact that the reference set does not contain the 'right' type of helical spectra which also raises difficulty for the peptide in trifluoroethanol (TFE) discussed below. It is also interesting to note that the α-helical structure gradually decreases from 45° to 75°, whereas the β-sheet structure increases over a much small temperature range (45–55°).

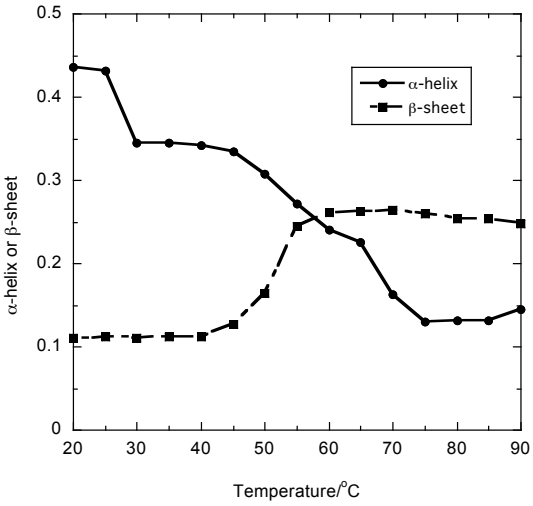
(a) (b)

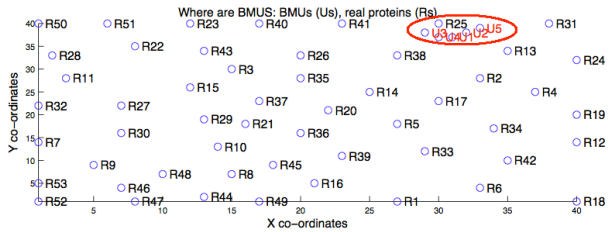

Δε
Increasing T
Increasing T
20°
25
30
35
40
45
50
55
60
65
70
75
80
85
90
Wavelength (nm)

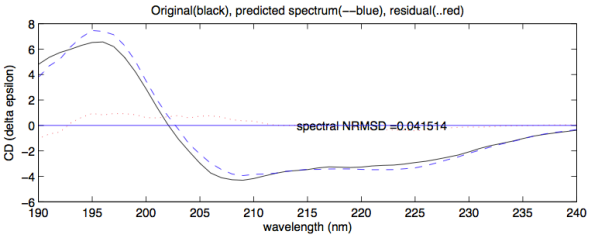
(c)


Spectral NRMSD
20oC
25
30
35
40
45
50
55
60
65
70
75
80
85
90
Q
Scaling factor

(d)


α-helix
β-sheet
α-helix or β-sheet
Temperature/°C


Where are BMUS: BMUs (Us), real proteins (Rs)
Y co−ordinates
X co−ordinates


Original(black), predicted spectrum(---blue), residual(..red)
spectral NRMSD =0.041514
CD (delta epsilon)
wavelength (nm)

(e)

(f)

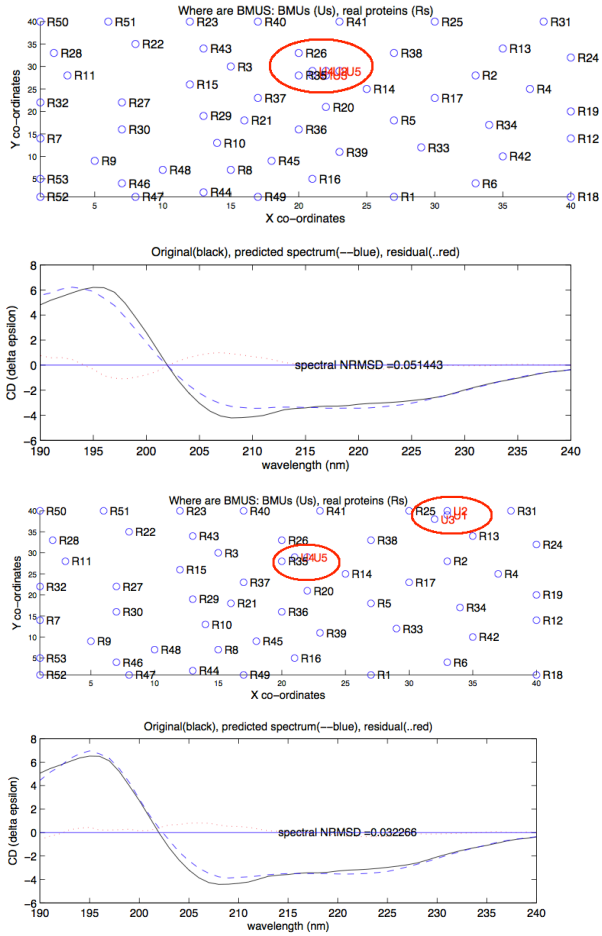


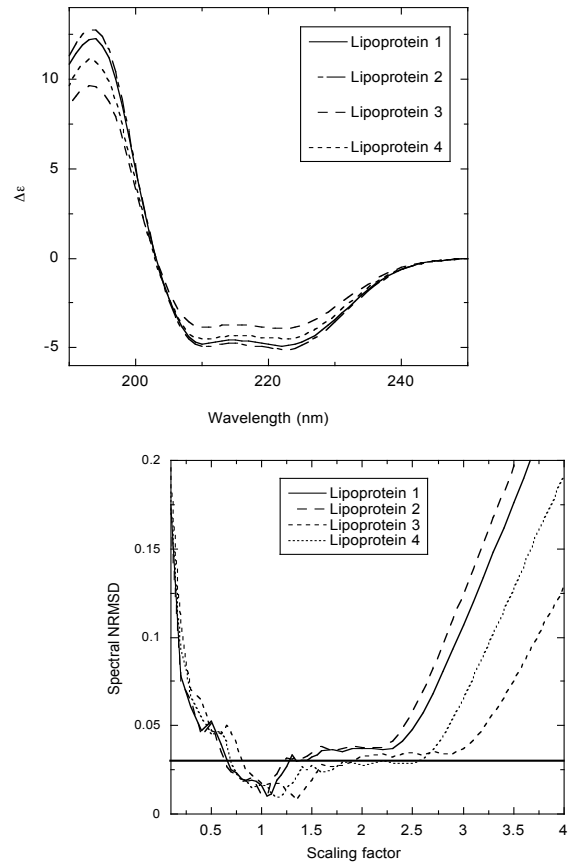
Figure 5: (a) CD spectra of insulin (pH 2.4) as a function of temperature (assuming nominal concentration 0.1 mg/mL in 1 mm path length cuvette). (b) NRMSD for SSNN output. Plots are vertically displaced by 0.01 for each temperature increment, with 20°C at the bottom. (c) SSNN α-helix and β-sheet as a function of temperature, using scaling factor 2 for 20–70° and 1.8 for the remainder. SSNN output for (d) 25° C with concentration scaling factor 2.0 (36% helix), (e) 30° C with concentration scaling factor 2.0 (34% helix), and (f) 30° C with concentration scaling factor 2.1 (41% helix).

Lipoproteins

CD spectroscopists frequently use protein concentrations of ~0.1 mg/mL, which for a pure protein corresponds to ~ 910–950 μM amino acid residue concentration which gives a good far UV (*i.e.* amide chromophore) CD spectrum in a 1 mm cuvette (as long as the buffer does not absorb light significantly). However, by definition, lipoproteins include lipids which are invisible in the spectrum but contribute to the mass and often affect protein concentration determination methods. Figure 6a shows the overlay of the CD spectra of 4 high density lipoproteins (L1–L4), which were all thought to be 0.1 mg/mL in concentration. Literature (*e.g.* [10]) led us to expect helical content when folded of between 50% and 80%. Some of our spectra are almost identical in spectral shape, but differ in magnitude. These spectra had been collected in our laboratory and abandoned, as the results could not be interpreted with available structure fitting methodologies.

Plots of spectral NRMSDs versus concentration scaling factor for L1–L4 are shown in Figure 6b. The NRMSD minima are for scaling factors: 1.05, 1.05, 1.35, and 1.2 for L1 to L4 respectively. The different predictions for neighbouring scaling factors give an indication of the percentage errors in the fits. Thus we conclude that the original protein concentrations were respectively 0.093, 0.095, 0.074, 0.083 mg/mL (within ±5%). Figure 6c illustrates the quality of the model spectrum overlaid with the original for L4 (the worst fit of the 4 proteins). L1 and L3 are 63% helix, 6–7% sheet and L2 and L4 are 65% helix, 6% sheet with errors (as determined by nearest scaling factor fits) of ~3% for the helices and 2% for the sheets.

(a) (b)



(c)

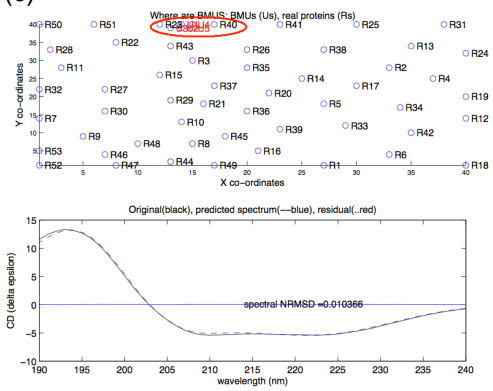


Figure 6: (a) CD spectra of 4 lipoproteins, L1–L4, converted to Δε assuming original protein concentration was 0.1 mg/mL in a 1 mm path length cuvette and the average amino acid molecular mass was 105 *u.* (b) SSNN3-multiple NRMSD for the 4 proteins versus concentration scaling factor. (c) SSNN3 output for L4 with concentration scaling factor 1.2.

ZapA: wild type and mutants

Escherichia Coli ZapA is a 104-residue protein which binds to the cell division protein FtsZ.[15,16,17] We had been interested in whether the wild type (WT) protein's structure changed when key residues (denoted *e.g.* I83E) were mutated. Although we had collected CD data (Figure 7a), our analysis had been hindered by very inaccurate concentration determinations. We therefore implemented SSNN and plotted the NRMSD versus concentration scaling factor. The answer is perhaps less clear than in the previous example with the NRMSD curves oscillating. However, this is partly the illusion of a different scale and partly that different local spectral NRMSD minima optimize different parts of the spectrum. Minima are at respectively 9 or 9.8, 3.6, and 4.4 for WT, I83E and V17E mutants. The last of these is illustrated in Figure 7c. The structure vectors for the two WT factors are (0.42,0.20,0.028,0.023,0.12,0.21) and 0.45,0.21,0.013591,0.017,0.11,0.20) with factor 9.8 being perhaps a slightly better visual fit

than 9 for the WT. We take the WT α-helical content to be (64±2)%. By way of contrast I83E has structure vector (0.33,0.19,0.047,0.043,0.14,0.25) so only 52% helix and V17E has vector (0.45,0.21,0.013,0.017,0.10,0.21) so 66% helix.

Even allowing for the uncertainties suggested in the predictions, I83E is significantly less helical than the other two proteins. According to the *Pseudomonas aeruginosa* crystal structure [17] we would expect ~ 64% helix and ~10% sheet for the WT ZapA. This helical content compares very well with the above SSNN results for WT and V17E (though the sheet predictions are low suggesting solution flexibility at the ends). However, I83E has significantly less helix predicted. Even allowing for errors predictions, we conclude that the I83E mutation disrupts some of the helical character of ZapA but the V17E mutant does not. Consistent with this is the fact that position 83 is in the coiled-coil region of the protein whereas position 17 is at the end of strand 2.[16, 17]

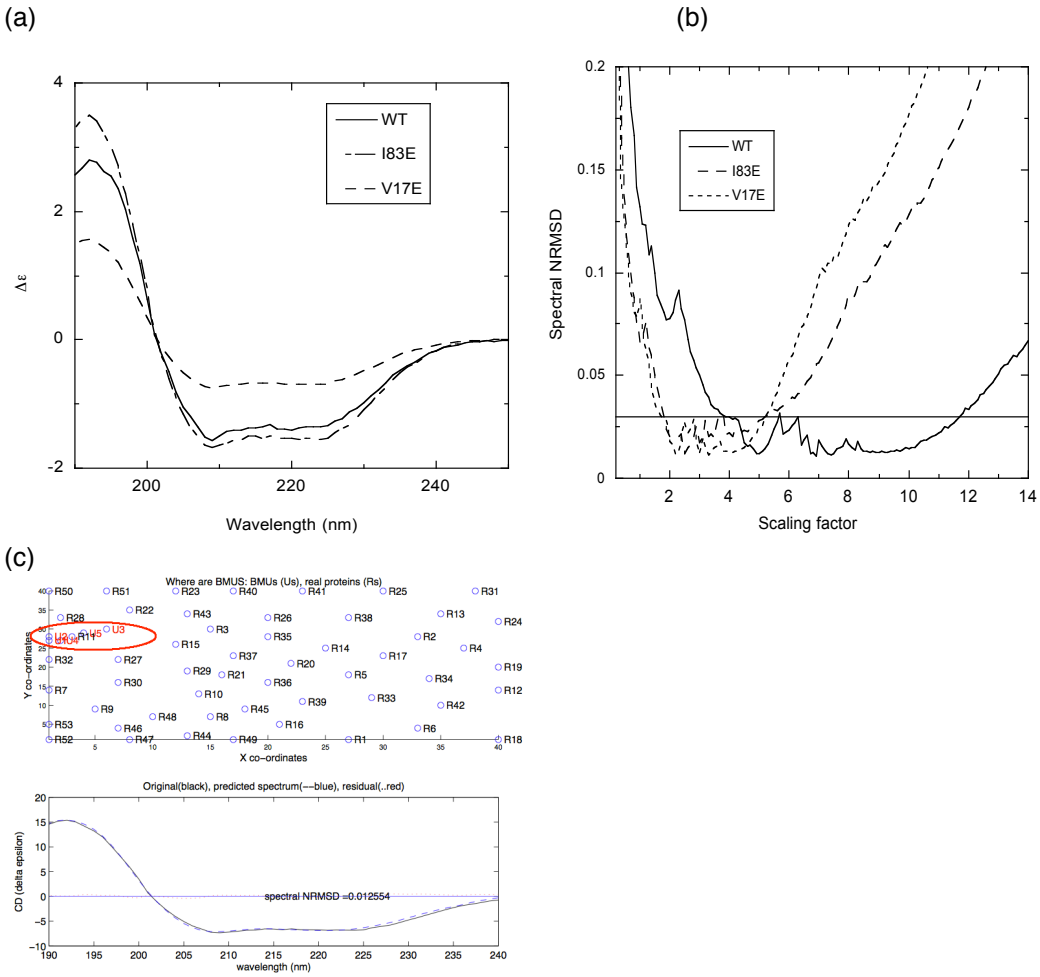


Figure 7: CD spectra of ZapA. (a) WT, I83E and V17E with Δε determined assuming nominal concentration 0.1 mg/mL in 1 mm pathlength cuvette and the average amino acid molecular mass 116 *u*. (b) SSNN3-multiple NRMSD for the 3 proteins versus concentration scaling factor with 0.03 quality indicator shown. (c) Overlay of experiment and model spectra for scaling factors = 4.4 for V17E.

Toxins

CD data were collected for a series of related bacterial proteins (toxins) as shown in Figure 8a. All samples had been dialysed to remove high concentrations of salt that interfered with CD data collection resulting in unknown concentrations. SSNNGUI was implemented with a range of scaling factors. Toxins 2 and 3 have spectra of almost the same shape, but the poor quality of the spectra at low wavelength result in different structure predictions, respectively 29% helix and 20% sheet for toxin 2 with scaling factor 0.6 and 32% helix and 12% sheet for toxin 3 with

scaling factor 1.1. No reasonable fit emerged for Toxin 1 which led us to re-examine the raw CD data files, which showed HT voltages above 600 V below 200 nm for Toxin 1 and below 196 nm for the others. So the short wavelength data were at fault not the fitting methodology. We therefore prepared new protein samples, which showed that the original spectra were attenuation at low wavelength. The typical minimum NRMSD was 0.016 with 35% α-helix and 12% β-sheet.

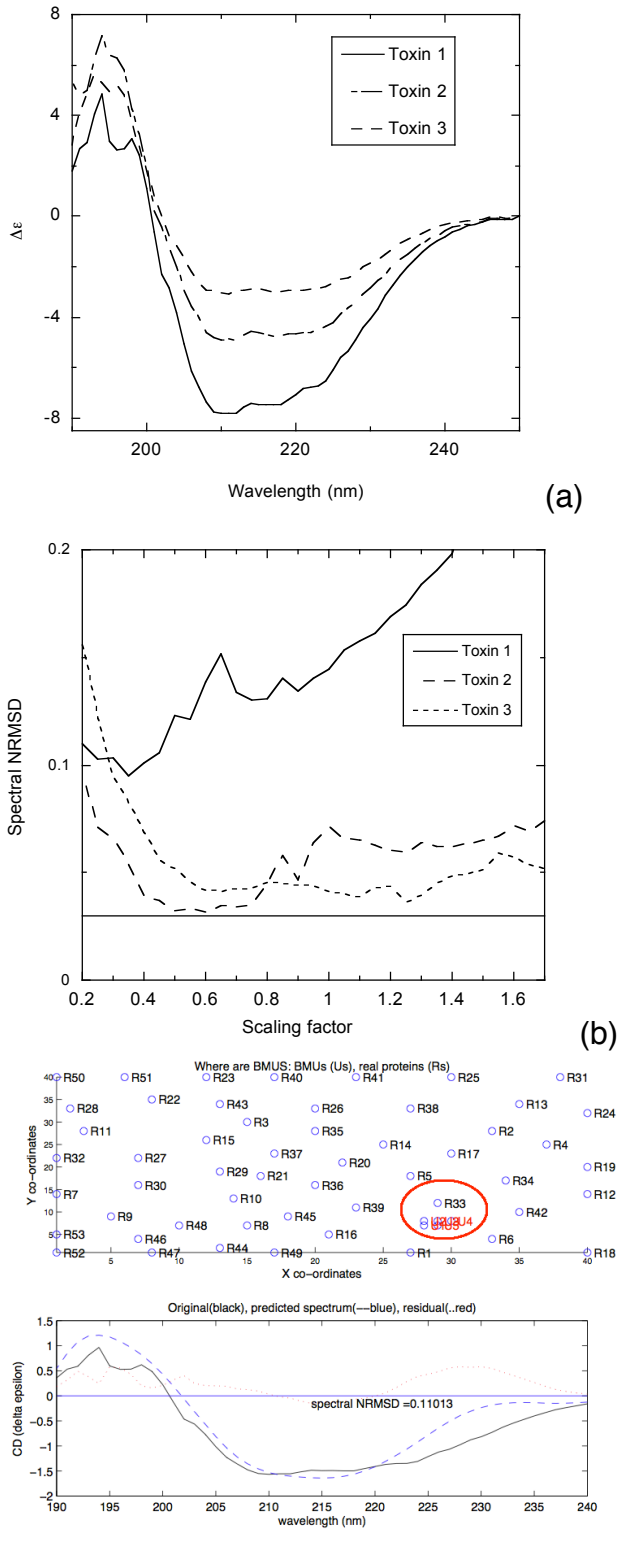


Figure 8: (a) Toxin CD spectra with Δε determined assuming nominal concentration 0.1 mg/mL in 1 mm pathlength cuvette and the average amino acid molecular mass 105 *u*. (b) SSNN3-multiple NRMSD for the 3 proteins versus concentration scaling factor with 0.03 quality indicator shown. (c) Overlay of experiment and "best fit" (see text for comment) model spectra for Toxin 1, scaling factor = 0.2.)

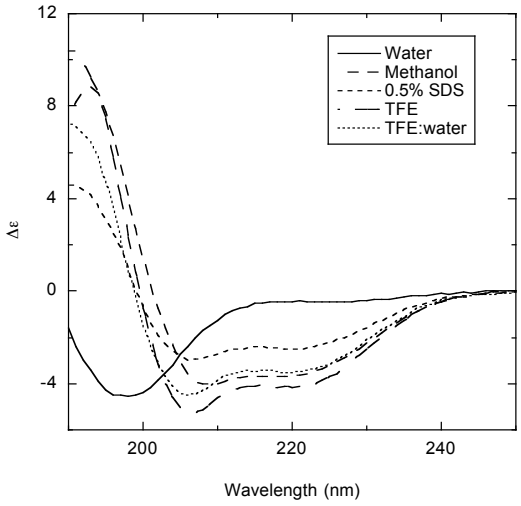
Peptide structure fitting

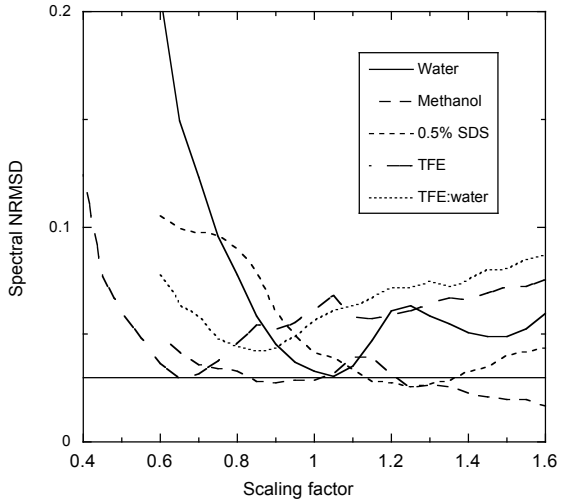
Peptides are a challenge to structure fitting programs as they tend to adopt a single secondary structure motif with frayed ends. Further, any sample may be a mixture of populations. We wished to see whether SSNN could at least be used to rank relative amounts of folding for peptides. Initial attempts to use SSNN with CDDATA.48 resulted in BMUs at the very edge of the maps and poor fits for unfolded peptides and for peptides suspected to be well-folded helices. With hindsight this should not have been surprising given CDDATA.48 contained data only from globular proteins which all have a mix of secondary structure motifs. To produce better peptide structure estimates we enhanced CDDATA.48 with 5 more spectra. Three spectra to mimic unfolded peptides were included: MSLSRRQAAQASGIALCAGAVPLKASA in water taken from reference,[18] and the spectra for N-formyl acetic acid and N-acetyl valine,[19] which have 100% 'random coil' structure. Two more reference spectra were constructed by taking the spectrum for myoglobin (from CDDATA.48) and for a helical aurein peptide [20] and scaling them to have the accepted maximum magnitude value of –13 $mol^{-1}cm^{-1}dm^3$ [21] at 222 or 208 nm. Scaled-myoglobin and the aurein were taken to have 100 % α-helix structure.

To test the usefulness of SSNN with the enhanced reference set for assessing secondary structures of peptides we used the data from a systematic study of a 27-mer SufI signal peptide from the *Escherichia coli* Tat system [15] with sequence: MSLSRRQFIQASGIALCAGAVPLKASA. Here we consider the peptide structure in water, methanol, 0.05% SDS, TFE, and TFE:water (Figure 9a). The overlay of the NRMSD plots versus concentration factor in Figure 9b suggest that the nominal concentration of 0.1 mg/mL is fairly close to the true value for all solvents except TFE (see below). The water model spectrum for scaling factor 1.05 (Figure 9c), gives a good fit to experiment letting us conclude that the prediction of only a small amount of secondary structure (5% helix, 9% sheet) is correct. One of the BMUs for this fit is spectrum 49, one of our additions to the training set. The structure predictions for the neighbouring scaling factors suggest ~ 2% error.

The methanol NRMSD plot (Figure 9b) is fairly flat, with multiple minima. The factor 1.25 fit looks best (Figure 9d) suggesting methanol induces ~51% helix and ~10% sheet. The factor 1.0 vector (0.30,0.18,0.070,0.055,0.12,0.29) leads us to conclude that in this case error is of the order of 5%. The other three solvents all have reasonable, but not good, fits at their NRMSD minima with helix/sheet percentages being 32%/17% for SDS (factor 1.2), 36%/16% for TFE factor (factor 0.65), and 33%/16% for TFE:water (factor 0.85). To confirm these values we would suggest some titration experiments with different mixed solvents. It is hard to imagine how the TFE concentration has been underestimated by 35% since the sample was prepared using a micro-balance. The local minimum near scaling factor 1.0 suggests 54% helix and 10% sheet. We would expect TFE to be at least as effective as MeOH in inducing helical structure, so we conclude that the quality of the fit is suffering from a lack of reference spectra where 208 nm is significantly larger in magnitude than 222 nm as was discussed above for insulin.

(a) (b)


12
8
4
0
-4
Δε
200
220
240
Wavelength (nm)
Water
Methanol
0.5% SDS
TFE
TFE:water


0.2
0.1
0
Spectral NRMSD
0.4
0.6
0.8
1
1.2
1.4
1.6
Scaling factor
Water
Methanol
0.5% SDS
TFE
TFE:water

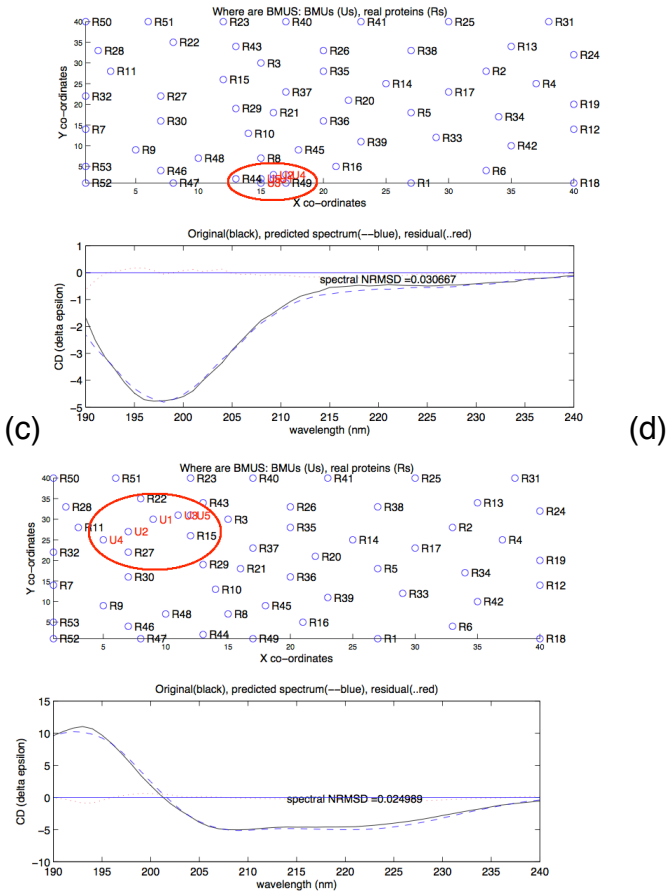


Figure 9: (a) CD spectra (with conversion to Δε per amino acid performed by assuming a concentration of 0.1 mg/mL in a 1 mm path length cuvette) of SufI peptide in different solvents. Data taken from reference [18]. (b) SSNN3-multiple NRMSD for SufI in different solvents versus concentration scaling factor with 0.03 quality indicator shown. (c) Overlay of experiment and best fit model spectrum for SufI in water [scaling factor = 1.05, structure vector (0.02,0.03,0.056,0.036,0.06,0.80)]. (d) Overlay of experiment and best fit model spectrum for SufI in methanol [scaling factor = 1.25, structure vector: (0.32,0.19,0.051,0.046,0.15,0.25)].

On the basis of this peptide example, we can conclude that when used with care, inspecting the fits, and considering the location of the BMUs on the map, SSNN can be a useful tool for peptides. In the original analyses of reference [18] it was assumed that the peptide could only adopt a helical or a random coil structure and percentages were determined using the CD magnitude at 222 nm. The results of using SSNN indicate a more complicated structural landscape for this 27-mer, which is in accord with the small size (or lack of with TFE) of the dip in CD intensity at ~216 nm expected between the 208 nm and 222 nm negative maxima for a helical protein.

Conclusion

In this paper we have shown how SSNN can be used to provide fairly reliable secondary structure estimates for proteins, even when the concentration of the sample is not known, as long as the spectral NRMSD versus concentration scaling factor plots have a clear minimum. Our experiences as summarized in the results section suggests that for the size of spectrum vector used in this work (51 data points spaced at 1 nm intervals), an NRMSD<0.03 gives a reasonable structure prediction. Where an oscillating NRMSD versus concentration scaling factor is observed, this is usually due to the set of BMUs changing as a function of concentration scaling factor. SSNN-multiple can be useful in making it clear when neighbouring scaling factors have BMUs in different parts of the map, which indicates a larger

error in the structure estimates. In such a situation an educated eye may be able to discern the best fit. Sometimes a poor fit reflects inadequacies in the reference set used to train SSNN, other times it reflects deficiencies in the experimental data such short wavelength data being significantly attenuated due to low photon count. Secondary structure estimates for both insulin and peptides dissolved in TFE would benefit from additions to the reference set. In some cases the results may suggest one of a small number of possibilities and an alternative technique such as infrared absorbance, Raman or Raman optical activity may be required to select between them. A series of spectra as a function of temperature or solvent competition may help clarify the appropriate scaling factor.

Most of the work reported in this paper has been performed with a GUI for the final module SSNN3-multiple. It takes as input the spectral SOM from SSNN1 and the structure map from SSNN2 as well as the experimental spectrum as a vector of 51 data points from 240 nm to 190 nm in 1 nm intervals. If a different wavelength range is required or a change in the reference data set is proposed, SSNN1 and SSNN2 will need to be rerun using the SSNN1_2.app. With the hindsight of the peptide work reported above, we have augmented CDDATA.48 reference set from CDPro to include 100% helical structures and 100% random coil structures thus extending the parameter space. The SSNNGUI that is now available at http://www2.warwick.ac.uk/fac/sci/chemistry/research/arodger/arodgergroup/research_intro/instrumentation/ssnn/ was trained with this larger reference set of 53 protein spectra. The fitting could be improved further by *e.g.* adding TFE-induced structures to a reference set. The lack of appropriate spectra in a reference set is usually apparent because the BMUs for a protein are very near the edge of the SOM rather than surrounded by other BMUs.

In looking for examples on which to test this concentration-optimising version of SSNN, we had many old data sets that had never been analysed once it had become apparent that we did not know the concentration. We are now able to proceed further with such data sets. A key to further progress will be enhancement of the training sets with particular classes of proteins, *e.g.* membrane proteins. Collecting the data for new spectra is one aspect of this, however, equally important is determination of the structure vector. In general, SSNN-multiple structures estimates for highly α-helical proteins are within about 3% and for highly β-sheet proteins have a low error despite the fact that concentration estimates may have high error. SSNN errors for mixed helix/sheet proteins may be of the order of 10% if the spectra NRMSD magnitudes are high (above 0.03).

Acknowledgements

VH thanks EPSRC for a PhD studentship through the MOAC Doctoral Training Centre grant number EP/F500378/1 and MS acknowledges funding from the FP7 Marie Curie Innovative Doctoral Training Programme.

References

1. Whitmore L, Wallace BA. DICHROWEB, an online server for protein secondary structure analyses from circular dichroism spectroscopic data. Nucleic Acids Res 2004;32(Web Server issue):W668-73.
2. Whitmore L, Wallace, B.A. Protein secondary structure analysis from circular dichroism spectroscopy: methods and reference databases. Biopolymers 2007;89:392-400.
3. Sreerama N, Venyaminov, S.Y., Woody, R.W. Estimation of Protein Secondary Structure from Circular Dichroism Spectra: Comparison of CONTIN, SELCON, and CDSSTR Methods with an Expanded Reference Set. Anal. Biochem. 2000;287:252-260.

4. Sreerama N, Woody, R.W. A self-consistent method for the analysis of protein secondary structure from Circular dichroism. Analyt. Biochem. 1993;209:32-44.
5. Hall V, Nash, A., Hines, E., Rodger, A. Elucidating protein secondary structure with circular dichroism and a neural network. Journal of Computational Chemistry 2013;34(32):2774-2786.
6. Hall V, Nash, A., Rodger, A. SSNN, a method for neural network protein secondary structure fitting using circular dichroism data. Submitted to Analytical Methods.
7. Andrade MA, Chacon P, Merelo JJ, Moran F. Evaluation of Secondary Structure of Proteins from Uv Circular-Dichroism Spectra Using an Unsupervised Learning Neural-Network. Protein Engineering 1993;6(4):383-390.
8. Sreerama N, Venyaminov S.Y., Woody, R.W. Estimation of protein secondary structure from CD spectra: Inclusion of denatured proteins with native protein in the analysis. Anal. Biochem. 2000;287:243-251.
9. Lees JG, Miles AJ, Wien F, Wallace BA. A reference database for circular dichroism spectroscopy covering fold and secondary structure space. Bioinformatics 2006;22(16):1955-1962.
10. Sevugan Chetty P, Mayne L, Kan Z-Y, Lund-Katz S, Englander SW, Phillips MC. Apolipoprotein A-I helical structure and stability in discoidal high-density lipoprotein (HDL) particles by hydrogen exchange and mass spectrometry. Proceedings of the National Academy of Sciences 2012;109(29):11687-11692.
11. Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. . Biopolymers 1983;22:2577–2637.
12. Shibata A, Yamamoto M, Yamashita T, Chiou J-S, Kamaya H, Ueda I. Biphasic Effects of Alcohols on the Phase Transition of Poly(L-lysine) between α-Helix and β-Sheet Conformations. Biochemistry 1992;31:5728-5732.
13. Greenfield N, Fasman GD. Computed circular dichroism spectra for the evaluation of protein conformation. Biochemistry 1969;8:4108-4116.
14. Hua QX, Gozani SN, Chance RE, Hoffmann JA, Frank BH, Weiss MA. Structure of a protein in a kinetic trap. Nat. Struct. Biol. 1995;2:129-138.
15. Pacheco-Gomez R, Cheng X, Hicks MR, Smith CJ, Roper DI, Addinall S, Rodger A, Dafforn TR. Tetramerization of ZapA is required for FtsZ bundling. Biochem J;449(3):795-802.
16. Small E, Marrington, R., Roder, A, Dafforn, T.R. FtsZ polymer-bundling by the Escherichia coli ZapA orthologue, YgfE involves a conformational change in bound GTP. J. Mol. Biol. 2007;369:211-221.
17. Low H, Moncrieffe M, Lowe J. The Crystal Structure of ZapA and its Modulation of FtsZ Polymerisation. J. Mol. Biol. 2004;341:839-52.
18. Miguel MS, Marrington R, Rodger PM, Rodger A, Robinson C. An Escherichia coli twin-arginine signal peptide switches between helical and unstructured conformations depending on hydrophobicity of the environment. Eur. J. Biochem. 2003;270:3345-3352.
19. Gokce I, Woody RW, Anderluh G, Lakey JH. Single peptide bonds exhibit poly(pro)II (“random coil”) circular dichroism spectra. J. Am. Chem. Soc. 2005;127:9700-9701.

20. Nordén B, Rodger A, Dafforn TR. Linear dichroism and circular dichroism: a textbook on polarized spectroscopy. Cambridge: Royal Society of Chemistry; 2010. 304 p.
21. Berova N, Nakanishi K, Woody RW, editors. Circular dichroism principles and applications. 2nd ed. New York: Wiley-VCH; 2000.

Protein Secondary Structure Prediction from Circular Dichroism Spectra Using a Self-organising Map with Concentration Correction

VINCENT HALL, MEROPI SKLEPARI, AND ALISON RODGER

Supplementary Information

Lipoproteins

Table S I: The predicted structures of the lipoproteins tested with SSNN GUI for all concentration scaling factors.

Concentration 0.1	Alpha-regular	Alpha-disorted	Beta-regular	Beta-disorted	Turns	Other
lipoprotein 1	0.014262	0.048501	0.26341	0.14546	0.21261	0.31576
lipoprotein 2	0.014951	0.050093	0.27105	0.1451	0.21012	0.30869
lipoprotein 3	0.011942	0.044572	0.24141	0.14452	0.22426	0.3333
lipoprotein 4	0.013403	0.047352	0.25117	0.14488	0.21848	0.32471
0.2						
lipoprotein 1	0.020358	0.051455	0.29918	0.13834	0.19897	0.2917
lipoprotein 2	0.019757	0.049894	0.30153	0.13744	0.19732	0.29406
lipoprotein 3	0.017688	0.050781	0.2927	0.14634	0.204	0.28848
lipoprotein 4	0.021252	0.054455	0.2926	0.14385	0.20159	0.28625
0.3						
lipoprotein 1	0.067308	0.065269	0.23626	0.11614	0.22545	0.28957
lipoprotein 2	0.06796	0.065562	0.23521	0.11595	0.22564	0.28968
lipoprotein 3	0.021811	0.053098	0.30036	0.13664	0.19378	0.29431
lipoprotein 4	0.055284	0.058422	0.25864	0.12144	0.21613	0.29009
0.4						
lipoprotein 1	0.14383	0.1069	0.14308	0.094367	0.2336	0.27822
lipoprotein 2	0.14897	0.11058	0.1327	0.090714	0.23452	0.28252
lipoprotein 3	0.073032	0.068205	0.23091	0.11618	0.22537	0.28631
lipoprotein 4	0.09858	0.080816	0.19593	0.11053	0.23107	0.28307
0.5						
lipoprotein 1	0.17428	0.12475	0.092071	0.070912	0.24241	0.29557
lipoprotein 2	0.17184	0.1237	0.088547	0.06962	0.24655	0.29974
lipoprotein 3	0.14389	0.10694	0.14298	0.094328	0.23361	0.27826
lipoprotein 4	0.16013	0.11593	0.11677	0.083078	0.23597	0.28813

0.6						
lipoprotein 1	0.22546	0.14977	0.061583	0.06964	0.20377	0.28978
lipoprotein 2	0.25156	0.1591	0.060937	0.069562	0.18645	0.27239
lipoprotein 3	0.17002	0.12097	0.098439	0.074612	0.24233	0.29363
lipoprotein 4	0.18171	0.13431	0.072547	0.064502	0.23659	0.31035
0.7						
lipoprotein 1	0.27571	0.16441	0.062926	0.070872	0.17986	0.24623
lipoprotein 2	0.27677	0.16361	0.065516	0.071978	0.18355	0.23857
lipoprotein 3	0.21	0.14557	0.059672	0.066078	0.21518	0.3035
lipoprotein 4	0.25187	0.16064	0.060977	0.066953	0.18579	0.27377
0.8						
lipoprotein 1	0.29202	0.16921	0.067138	0.069558	0.17149	0.23059
lipoprotein 2	0.28679	0.16378	0.065518	0.069931	0.17914	0.23484
lipoprotein 3	0.25293	0.16077	0.061063	0.067187	0.18546	0.27258
lipoprotein 4	0.27606	0.16352	0.065724	0.071843	0.18382	0.23903
0.9						
lipoprotein 1	0.30118	0.16981	0.064452	0.063092	0.16517	0.2363
lipoprotein 2	0.32963	0.18861	0.062708	0.055574	0.14049	0.22298
lipoprotein 3	0.27829	0.16544	0.064438	0.071135	0.17962	0.24108
lipoprotein 4	0.29155	0.16886	0.067225	0.069332	0.17122	0.23182
1						
lipoprotein 1	0.37307	0.20784	0.050736	0.038988	0.11205	0.21731
lipoprotein 2	0.42532	0.20314	0.037853	0.029121	0.11487	0.18968
lipoprotein 3	0.28681	0.16608	0.066914	0.070533	0.17891	0.23075
lipoprotein 4	0.30434	0.17227	0.064316	0.062091	0.16184	0.23514
1.2						
lipoprotein 1	0.53908	0.1739	0.018855	0.01366	0.0926	0.16191
lipoprotein 2	0.56027	0.1685	0.018697	0.013503	0.08762	0.15142
lipoprotein 3	0.35701	0.20381	0.056729	0.044064	0.11459	0.2238
lipoprotein 4	0.45056	0.19774	0.032536	0.025746	0.11188	0.18154
1.4						
lipoprotein 1	0.69528	0.12997	0.01162	0.0082634	0.051245	0.10361
lipoprotein 2	0.69609	0.13325	0.0094157	0.0066956	0.052767	0.10179
lipoprotein 3	0.47503	0.19334	0.027007	0.020731	0.10624	0.17765
lipoprotein 4	0.56242	0.16799	0.018627	0.013447	0.08708	0.15044
1.6						
lipoprotein 1	0.71046	0.14234	0	0	0.048521	0.098677
lipoprotein 2	0.70119	0.14387	0.0022733	0.0016166	0.047171	0.10388
lipoprotein 3	0.56202	0.16808	0.01864	0.013458	0.087182	0.15062

lipoprotein 4	0.69506	0.13455	0.0090069	0.0064049	0.052756	0.10222
1.8						
lipoprotein 1	0.73716	0.12163	0.0019883	0.0014139	0.047229	0.090583
lipoprotein 2	0.73841	0.12306	0.0020441	0.0014536	0.044092	0.090949
lipoprotein 3	0.69527	0.13011	0.011552	0.0082145	0.051245	0.10361
lipoprotein 4	0.70953	0.14551	0	0	0.044822	0.10014
2						
lipoprotein 1	0.76651	0.11466	0	0	0.036096	0.082733
lipoprotein 2	0.7819	0.10675	0	0	0.03419	0.077159
lipoprotein 3	0.71128	0.14155	0	0	0.049089	0.098082
lipoprotein 4	0.73838	0.12351	0.0019197	0.0013651	0.043868	0.090954
2.2						
lipoprotein 1	0.80901	0.090747	0	0	0.034375	0.065865
lipoprotein 2	0.82031	0.081754	0.0013303	0.00094597	0.033014	0.062647
lipoprotein 3	0.71537	0.13386	0.0023665	0.0016829	0.048638	0.098091
lipoprotein 4	0.76394	0.11741	0	0	0.034759	0.083887
2.4						
lipoprotein 1	0.83315	0.076987	0.0013513	0.0009609	0.029141	0.058407
lipoprotein 2	0.83846	0.075507	0.0013335	0.00094826	0.026992	0.05676
lipoprotein 3	0.74384	0.12174	0.0020509	0.0014584	0.040589	0.090327
lipoprotein 4	0.79663	0.098363	0	0	0.034764	0.070247
2.6						
lipoprotein 1	0.85301	0.067532	0.0013188	0.00093783	0.025482	0.051721
lipoprotein 2	0.85318	0.067473	0.001308	0.00093009	0.025454	0.051656
lipoprotein 3	0.78187	0.10675	0	0	0.034163	0.077215
lipoprotein 4	0.83316	0.076954	0.0013626	0.00096897	0.029145	0.058413
2.8						
lipoprotein 1	0.85089	0.069717	0.001305	0.00092801	0.02467	0.052495
lipoprotein 2	0.85089	0.069721	0.001302	0.00092585	0.024668	0.052491
lipoprotein 3	0.81962	0.0822	0.0012645	0.0008992	0.033181	0.062834
lipoprotein 4	0.85301	0.06753	0.0013192	0.00093808	0.025481	0.05172
3						
lipoprotein 1	0.85088	0.069729	0.0013023	0.00092604	0.02467	0.052496
lipoprotein 2	0.85086	0.069741	0.0013019	0.00092578	0.024671	0.052503
lipoprotein 3	0.83314	0.076984	0.001354	0.00096281	0.029145	0.058412
lipoprotein 4	0.85099	0.069674	0.0012994	0.00092402	0.02466	0.052457
3.2						
lipoprotein 1	0.85084	0.06975	0.0013027	0.00092633	0.024673	0.05251
lipoprotein 2	0.85081	0.069762	0.0013031	0.00092661	0.024675	0.052519

lipoprotein 3	0.8529	0.067571	0.0013251	0.0009423	0.025499	0.051761
lipoprotein 4	0.85097	0.069691	0.0012971	0.00092235	0.02466	0.052462
3.4						
lipoprotein 1	0.8508	0.069769	0.0013038	0.00092713	0.024677	0.052524
lipoprotein 2	0.85078	0.06978	0.0013044	0.00092755	0.024679	0.052532
lipoprotein 3	0.85089	0.069709	0.0013064	0.000929	0.024671	0.052493
lipoprotein 4	0.85091	0.06972	0.0012984	0.00092331	0.024665	0.052482
3.6						
lipoprotein 1	0.85077	0.069784	0.001305	0.00092797	0.02468	0.052536
lipoprotein 2	0.85075	0.069794	0.0013056	0.00092841	0.024682	0.052544
lipoprotein 3	0.85092	0.069705	0.001301	0.00092512	0.024665	0.052479
lipoprotein 4	0.85086	0.069745	0.0013003	0.00092465	0.02467	0.052502
3.8						
lipoprotein 1	0.85074	0.069797	0.0013061	0.00092873	0.024683	0.052547
lipoprotein 2	0.85072	0.069805	0.0013067	0.00092916	0.024685	0.052553
lipoprotein 3	0.8509	0.069722	0.0013004	0.00092472	0.024667	0.052488
lipoprotein 4	0.85082	0.069764	0.0013021	0.0009259	0.024675	0.052518
4						
lipoprotein 1	0.85072	0.069808	0.001307	0.00092941	0.024685	0.052555
lipoprotein 2	0.8507	0.069815	0.0013076	0.00092981	0.024687	0.052561
lipoprotein 3	0.85086	0.069741	0.0013012	0.00092531	0.024671	0.052502
lipoprotein 4	0.85078	0.06978	0.0013036	0.00092698	0.024679	0.052531

Table S II: Spectral NRMSD of lipoproteins for each concentration scaling factor (denoted ‘Co ncentration’) examined.

Conc.	Lipoprotein 1	Lipoprotein 2	Lipoprotein 3	Lipoprotein 4
0.1	0.17373	0.16101	0.19559	0.18199
0.2	0.078179	0.075448	0.11093	0.094671
0.3	0.061069	0.06393	0.070452	0.066127
0.4	0.04708	0.044042	0.064299	0.054296
0.5	0.052831	0.051135	0.047451	0.044787
0.6	0.039238	0.035442	0.045115	0.047038
0.7	0.025232	0.024039	0.045917	0.029046
0.8	0.01982	0.019678	0.029543	0.018716
0.9	0.019099	0.02085	0.021897	0.014965
1	0.01606	0.011649	0.01752	0.016609

1.2	0.021886	0.027913	0.017271	0.010366
1.4	0.030681	0.031658	0.012088	0.026781
1.6	0.035815	0.03795	0.028181	0.02395
1.8	0.035241	0.03721	0.027439	0.029177
2	0.036921	0.037531	0.032832	0.027642
2.2	0.03666	0.037839	0.033985	0.029618
2.4	0.041127	0.048554	0.033851	0.028793
2.6	0.057096	0.069767	0.033743	0.030886
2.8	0.080572	0.095972	0.034215	0.044315
3	0.10673	0.12403	0.037175	0.065682
3.2	0.13404	0.15289	0.048506	0.089628
3.4	0.16193	0.18217	0.066051	0.11459
3.6	0.19013	0.21171	0.086106	0.14002
3.8	0.21852	0.24139	0.10734	0.16571
4	0.24705	0.27119	0.12916	0.19155

ZapA

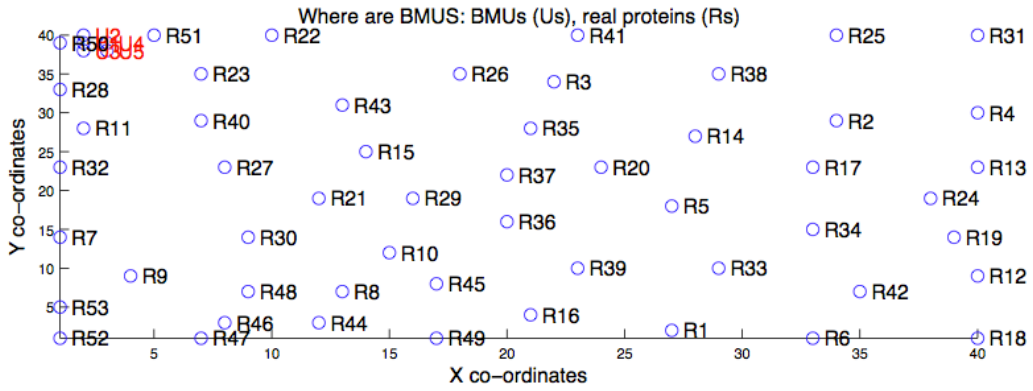


Figure S1: ZapA 3, wild type: locations of BMUs, black ones marked with 'R' are training set proteins, red ones marked with 'U' are the 5 BMUs that make up the model spectrum and the structure estimation. Here ZapA WT has a concentration scaling factor of 8.8 (see Figure S8 for further details).

See main text to see if this should be deleted, or replaced by new, correct stuff.

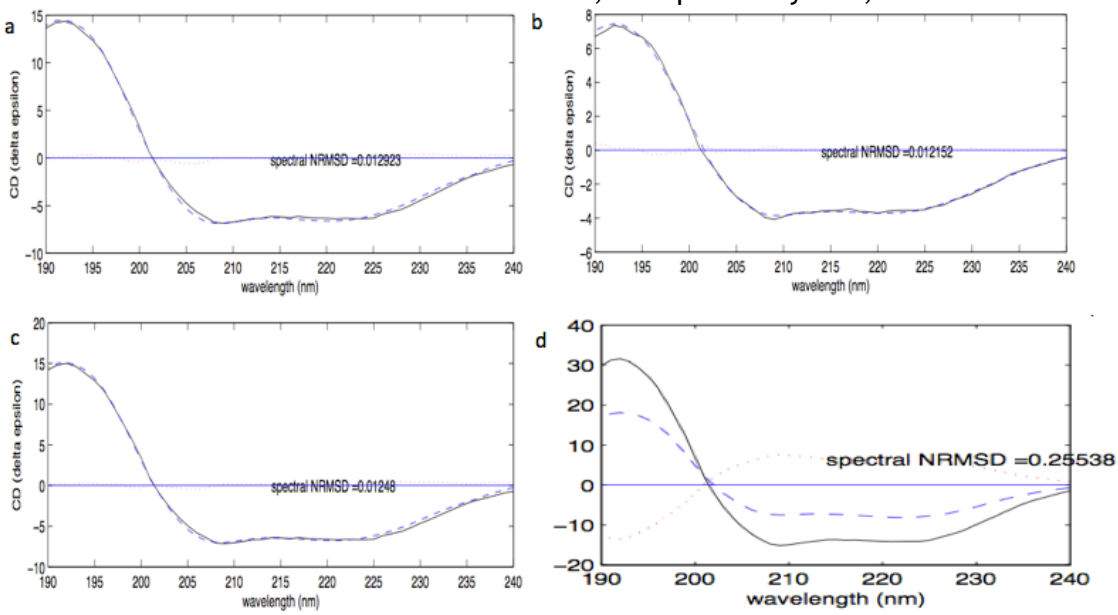


Figure S2: ZapA comparisons of SSNN model spectra with their respective original spectra scaled with best scaling factors (denoted 'Concentration'). The scaled experimental spectra are in black, the models are in dashed blue. (a) ZapA wild type scaled by 8.8, indicating an original concentration of 0.011 mg/mL. (b) ZapA I83E mutant scaled by 2.5 indicating an original concentration of 0.04 mg/mL. (c) ZapA V17E scaled by 4.1 indicating an original concentration of 0.024 mg/mL. (d) V17E scaled at 2.0 times original concentration, note the very large NRMSD.

Analytical Methods RSCPublishing

ARTICLE

SSNN, a method for neural network protein secondary structure fitting using circular dichroism data

Cite this: DOI: 10.1039/xoxxoooooo VINCENT HALL,[a,b] ANTHONY NASH,[a,b,c] ALISON RODGER*[b,c]

Received (in XXX, XXX) Xth XXXXXXXXX 2013, Accepted Xth XXXXXXXXX 20XX

Circular dichroism (CD) spectroscopy is a quick method for measuring data which can be used to determine the average secondary structures of proteins, probe their interactions with their environment, and aid in drug discovery. This paper describes the operation and testing of a self-organising map (SOM) structure-fitting methodology named Secondary Structure Neural Network (SSNN), which is a methodology for estimating protein secondary structure from CD spectra of unknown proteins using CD spectra of proteins with known X-ray structures. SSNN comes in two standalone MATLAB applications for estimating unknown proteins' structures, one that uses a pre-trained map and one that begins by training the SOM with a reference set of the user's choice. These are available at http://www2.warwick.ac.uk/fac/sci/chemistry/research/arodger/arodgergroup/research_intro/instrumentation/ssnn/ as SSNNGUI and SSNN1_2 respectively. They are available for both Macintosh and Windows formats with two reference sets: one obtained from the CDPro website, referred to as CDDATA.48 which has 48 protein spectra and structures, and one with 53 proteins (CDDATA.48 with 5 additional spectra). Here we compare SSNN with CDSSTR, a widely-used secondary structure methodology, and describe how to use the standalone SSNN applications. Current input format is $\Delta\varepsilon$ per amino acid residue from 240 nm to 190 nm in 1 nm steps for the known and unknown proteins and a vector summarising the secondary structure elements of the known proteins. The format is readily modified to include input data with *e.g.* extended wavelength ranges or different assignment of secondary structures.

Received ooth October 2013,
Accepted ooth January 2013

DOI: 10.1039/xoxxoooooo
www.rsc.org

Introduction

Circular dichroism spectroscopy is the difference in absorbance of left and right circularly polarized light. It is probably most often used to estimate the percentages of different secondary structures that are present in proteins. Assuming that high quality data have been collected for the sample of interest and the concentration is accurately known, then the question arises as to what method is best used to assign secondary structure motifs quantitatively. It is now generally recognised that the best approach is to use spectral data and secondary structures for an extensive reference set of proteins and then implement a process to estimate the secondary structure content of the unknown protein.[1-6]

Some CD spectral features are readily apparent, for example, an α-helix is characterised by a large positive band at 190 nm (part of a $\pi\rightarrow\pi^*$ exciton couplet), and two smaller negative bands at 208 nm (the other $\pi\rightarrow\pi^*$ component) and 222 nm ($n\rightarrow\pi^*$). β-sheets usually show a positive peak between 195 nm and 202 nm, and a negative signal between 215 nm and 220 nm, though sometimes resemble the 'random coil' and poly-proline II spectra which have a negative signal at 200 nm.[4] It is now widely accepted that this 200 nm negative band is dominated by contributions from residues with poly(proline) type-II conformations.[7] β-turns have a large negative band at 180 nm–190 nm, a positive signal in the 200 nm–205 nm range ($\pi\rightarrow\pi^*$), and a negative signal at 225 nm ($n\rightarrow\pi^*$).

A number of secondary structure analysis programs, based either in statistical methods or intelligent systems, exist that make quantitative assignments of percentages of structure type.[*e.g.* 1, 8-11] It is possible to make use of some of these on Dichroweb, an online server hosted at Birkbeck, University of London.[12] The commonly used statistical packages which are available on Dichroweb include: CONTIN which is a ridge regression

technique; CDSSTR (an update of 'VARSLC') which is a variable selection, or feature selection, method; and SELCON (now SELCON3) which is a self-consistent method together with a singular value decomposition, SVD, algorithm. Dichroweb includes one intelligent system approach, called K2d,[10] which is a self organizing map (SOM) neural network approach. Although the intelligent systems approach appears to have many advantages, K2d and its successors including K2D2/3 [13-15] and SOMCD,[16] have not been widely adopted by the CD community. As these methods are only available as pre-trained SOMs where the reference set and structural categories have been defined by the original authors, we chose to develop our own CD structure fitting SOM: Secondary Structure Neural Network (SSNN) so we could test it back-to-back with statistical methods and enable any user to train it with new spectral reference sets and different structure assignment methodologies if the researcher wished to do so. This is timely as new data bases are currently being developed, greatly facilitated by the recently established Protein Circular Dichroism Data Bank.[17]

In summary, SSNN has three independent modules that operate in sequence.

SSNN1: takes spectra for a set of proteins of known secondary structure content (the reference set) and trains (organises) them so that related spectra are put near each other on the map. The map has many more nodes to put spectra in than there are spectra, so the gaps are filled-in with intermediate, virtual spectra. These virtual spectra are made by taking weighted sums of the nearby experimentally-obtained spectra.

SSNN2: puts vectors of the secondary structure contents of the reference proteins onto a structures map that matches the output of SSNN1, with structure vectors created for the virtual spectra by using the same weighting for the virtual nodes as used for the spectra.

SSNN3: takes as input a CD spectrum of a structurally unknown protein (currently in units of $\Delta\varepsilon/(\mathrm{mol}^{-1}\mathrm{dm}^3\mathrm{cm}^{-1})$, where the concentration is that of amino acid residues rather than molecules of protein) and produces as output an estimate of its secondary structure, a model spectrum, and the spectral NRMSD (normalised root mean squared deviation) defined as

$$NRMSD = \frac{\sqrt{\left(\frac{\sum_i \left(x_{i,experiment} - x_{i,model}\right)^2}{N}\right)}}{M - m}$$

where x_i is the value at each wavelength (or structure), N is the number of data points, M is the largest intensity, and m is the smallest, so $(M–m)$ is the range. SSNN1 and SSNN2 need only be performed once for a given reference set.

In a previous paper [11] we determined the parameters required to optimize the performance of SSNN and showed that it compared well with SELCON3, K2d, and SOMCD in a leave-one-out comparison using CDDATA.48 from the CDPro web site (http://lamar.colostate.edu/~sreeram/CDPro/). At this time we made SSNN3 available pre-trained but without detailed instructions for use. The overall performance of SSNN-47 and SELCON3-47 is similar: SELCON3-47 "won" for 23 out of 48 spectra, being slightly better on average for α-helix and Other structure estimates. Whereas SSNN-47 "won" for 25 out of 48 spectra and on average was slightly better for mixed α-helix/β-sheets, and β-sheets and turns. Comparisons between SSNN and K2d and SOMCD were hampered by lack of re-trainable executable versions of K2D and SOMCD so we worked with what was available in Dichroweb [4] and the literature [16] creating a comparison methodology that dis-favoured SSNN. In summary, SSNN out-performed K2d and performed better than SOMCD for 22 of the 33 proteins for which results were available in reference [16].

The aim of this paper is to provide other CD users, including those in the biopharmaceutical industry, with the tools required to use all three modules of SSNN, thus enabling them to work with new reference sets and structure definitions should they so desire. We also show that SSNN has a firm place in the tool-kit for CD structure analysis by showing that it compares very well with CDSSTR, which Woody and Sreerama previously showed was better than SELCON3 for β-distorted structures.[3]

Methods

Reference data set

The reference data sets used by us to train SSNN in this work are CDDATA.48 (with data from 240 nm–190 nm taken from the CDPro website: http://lamar.colostate.edu/~sreeram/CDPro/) and CDDATA.48+5 which is CDDATA.48 augmented with 5 additional spectra, 2 representing 100% α-helix and 3 100% Other structures (some of these are extrapolations). The spectra are given in per residue molar absorbance units ($\Delta\varepsilon$= mol^{-1} dm^3 cm^{-1}) and the structures are assigned to 6 structure categories. This is the largest available CD reference set that has been consistently annotated with secondary structures. Each member of the input reference set has 57 numbers. In order they are: 51 for the spectral intensity at each wavelength from 240 nm to 190 nm, and 6 for the structure values in the order: (α-helix regular, α-helix distorted, β-sheet regular, β-sheet distorted, turns, Other structures). The CDPro website and its references explain the structure types which are summarised in reference [3]. In summary the 'regular' structures are the middle of helices and sheets and the 'distorted' structures at the ends. In the leave-one-out tests reported in this paper, each spectrum was removed from the reference set in turn and used as the test spectrum. CDSSTR or SSNN1 and SSNN2 was/were run with the resulting 47-member reference set. This was repeated for each member of the reference set, this is called leave-one-out cross-validation.

SSNNGUI

To run SSNN3, or "SSNNGUI", pretrained with CDDATA.48+5 file, the correct version for the computer being used should be downloaded from http://www2.warwick.ac.uk/fac/sci/chemistry/research/arodger/arodgergroup/research_intro/instrumentation/ssnn/.

The details of how to install and run SSNNGUI are in the supplementary information. What follows below is an outline of how to proceed and the results that will be obtained. Table 1 summarises the steps once the program has been installed on your computer. Figure 1 shows SSNNGUI_versA, the version of SSNN3 that does not need SSNN1 and SSNN2 to be trained before it is used as it already includes the results of training with

CADATA.48+5.

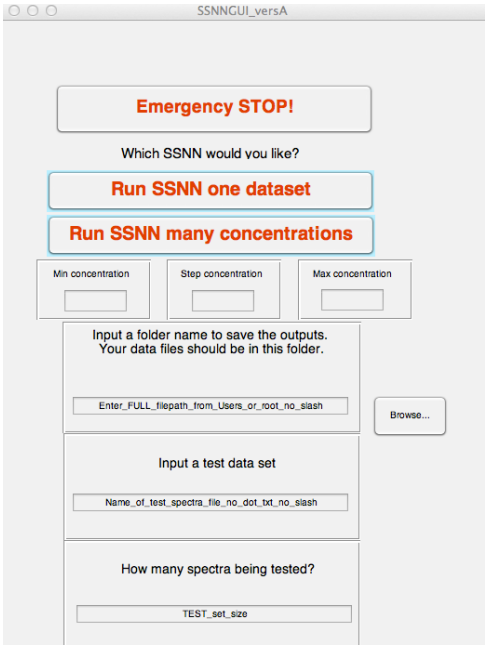


Figure 1: Screenshot of SSNNGUI 2013a.

Table 1. Protocol to run SSNN3 for test proteins of known concentration once the application has been installed (see Methods and Supplementary Information for details).

1. Navigate to the folder containing SSNNGUI_2013a.
2. Click the SSNNGUI.app icon or run the .sh file using Terminal as described in the Supplementary Information.
3. Click on the browse button beside the field that says <Input a folder name to save the outputs...>. Navigate to the folder containing < SSNNGUI_2013a>, and select *any* file in it (you may need to change the selection criterion to <All files>), as long as the folder is correct. The text in the SSNNGUI box will not change.
4. In the field called <Input a test data set> type the name of the test file, for example <toxins experiment 1 Friday>. The file type should be .txt, but do not include <.txt> in the file name. The test file should have columns of 51 data points representing the CD spectra from 240 nm to 190 nm. Multiple spectra can be in the same .txt file as tab or comma delimited columns. No other text should be present. See the example file included in the package.
5. In the field <How many spectra being tested?> put the number of spectra in that test file.
6. Hit the <Run SSNN one dataset> button ("one dataset" means one file with, perhaps, multiple spectra). In a few seconds one should be presented with the same number of windows as the number entered in step 5. These windows will contain two plots. The top plot will be the locations of all of the best matching units (BMUs), the bottom plot will be the experimental spectrum of a protein compared with the model of it that SSNN made, and a residual (the difference between the original spectrum and the model spectrum). The txt output files should also appear in an output folder labelled with the input data file name.

SSNN1_2

To make SSNN future compatible we have made SSNN available as a re-trainable package. The user will be able to use SSNN with different data sets as they become available. In the supplementary information is a guide on how to use the incarnation of SSNN that can be retrained.

RESULTS AND DISCUSSION

The main result of this paper is the production of a stand-alone pretrained GUI for SSNN, called SSNNGUI. We have also made versions of SSNN1, 2, and 3 all together in one application denoted SSNN1_2. They are available at http://www2.warwick.ac.uk/fac/sci/chemistry/research/arodger/arodgergroup/research_intro/instrumentation/ssnn/.

SSNN1_2 will allow CD practitioners to train the methodology with their own desired reference sets. They may then estimate CD secondary structures, with the confidence of having used their own reference spectra of the test proteins. Preferred reference sets might include additional proteins, different methods of structural annotation (including number of structure types), different wavelength ranges *etc.* This is particularly attractive as more good spectra and structures become available from CD, X-ray crystallography, NMR and other sources.

The SSNN application has various parameters that can be varied to train the program in a way tailored to the reference set used. For example, practitioners might like to change the number of iterations, as SSNN may take longer to learn a larger reference set than the 28,000 iterations that we found was best for about 50 spectra.[11]

Care should be taken in changing parameters. For example, the initial learning rate follows a negative exponential curve throughout the training process. Changing this rate will have an impact on the whole training. Making the initial learning rate too low will cause spectra map spectra to approximate the reference set (experimental) spectra too slowly. Making it too high will move spectra into position too quickly, which might not produce a broad enough region for each type of structure. Our experiments showed us learning too quickly can be a bad thing.[11] This may also depend on the size of the map.

The emergency stop button has been included in the program, though in our experience it has never had to be used in SSNNGUI in practice. It does shut down the application completely, although it takes a little time to do so.

Among other functions, SSNN model spectra residuals should also highlight which wavelengths of the test protein spectra produce the most error. SSNN treats all wavelengths equally, so a larger error in one region of a spectrum could throw off the estimation. If this correlates with a region of the spectrum where the data quality is poor, the user may want to ignore this contribution to the NRMSD. Users should also check that model spectra have appropriate intensities at certain wavelengths. For example β-sheet-rich protein models should have a single negative peak between 215 nm and 220 nm, whereas α-helix-rich proteins should have negative peaks at 208 nm and 222 nm.

Comparison of SSNN and CDSSTR

We previously ran SSNN1,2, and 3 48 times in a leave-one-out methodology using 47 spectra of CDDATA.48 as the reference data set and the omitted one as the test spectrum each time.[11] We refer to this as SSNN-47. In this way we ensured that the test spectra spanned structural space and also that the structure annotation was the same for the reference set and the test spectra. Table 2 and Table SI1 show net the results (final rows) of the previous work and details of an analogous leave-one-out test performed here for CDSSTR (CDSSTR-47) using the executable version available on the CDPro web site. The sum of the absolute values of the deviations from the fractions of real structures is also given for SELCON3-47 in the final lines of the Table SI1. SELCON3-47 data are from [11]. We also ran the leave-one-out test for SSNN-52 (*i.e.* CDDATA.48+5) and have shown the detailed results in Table SI1.

In summary, for the CDDATA.48 SELCON3 is best for high α-helix structures, SSNN is best for medium α-helix and 'Other' structures, and joint best for β-sheets. CDSSTR is joint best for distorted β-sheets. When reduced to 3 structure types α-helix, β-sheet and Turns-plus-other (data not shown), SELCON3 remains best for α-helices and SSNN for β-sheets and Turns-plus-Other. The somewhat disconcerting result given in Table SI1 is that CDSSTR has the best *spectral* NRMSD (Table SI1) but this does not translate into the best structural predictions. It is satisfying to note that the augmented reference set, CDDATA.48+5, leads to the SSNN-52 results being better than all the CDDATA.48 results.

As noted previously,[11] it is important to inspect the model spectrum outputted by a fitting program. The metrics currently used to assess fit do not give extra significance to the 218 nm region of the spectrum which is a negative maximum for β-sheet structures and a negative minimum between two negative maxima for α-helical structures. This is illustrated for carboxypeptidase A which has low spectral NRMSDs for CDSSTR-47 of 0.075 and for SSNN-47 of 0.067, but high structural NRMSDs of 0.68 and 0.34 respectively. The overlay of their plots is given in Figure 2a, showing the difference in spectral structure that is 'obvious' to the eye but not to an equally weighted numeric estimate of spectral NRMSD that the SSNN model spectrum is not helical enough. Conversely Rat Intestinal Fatty Acid Binding Protein (Figure 2b) is modelled to be more helix than reality. CDSSTR has much larger structural NRMSD than SSNN but the spectral fit looks better, illustrating the comment made above about CDSSTR.

Another example is provided by rat intestinal fatty acid binding protein, where both CDSSTR-47 and SSNN-47 make reasonably good spectral models (NRMSDs: 0.065, and 0.076 for CDSSTR-47 and SSNN-47 respectively), but very bad structural estimations (NRMSDs: 0.81 and 0.76). This indicated by poor agreement between model and experimental spectra at 218 nm, due in this case to the fact that this highly β-sheet spectrum has unusually large magnitude which is not reflected in the reference set (Figure 2b).

Table 2 shows the performance of the programs on particular classes of proteins. Mixed α/β proteins remain the most challenging for all the programs and a critical human eye is an invaluable tool for assessing the output from any of the programs. Although SSNN-47 and CDSSTR-47 mixed α/β proteins estimates are done as well as certain other structure types, mixed α/β proteins are consistently badly done by all methodologies.

The challenge in estimating the Other class is that it is an amalgam of various undefined structures that have different spectral features. Putting these into one class implies they have similar spectra and structures—which is misleading.

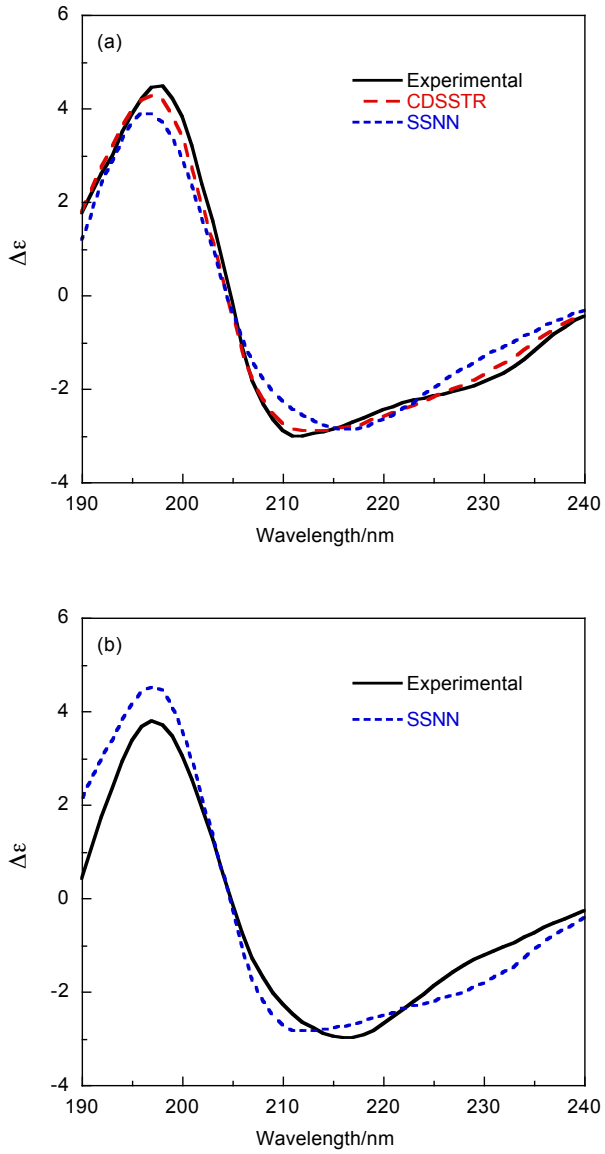


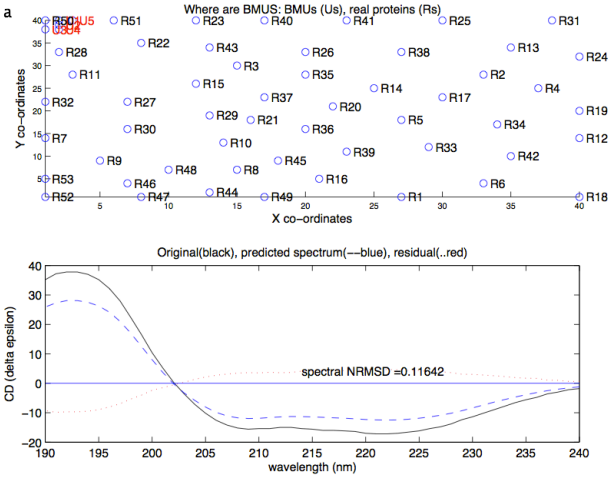
Figure 2. (a) Carboxypeptidase A experimental CD spectrum and best fit from CDSSTR-47 and SSNN-47. (b) Rat Intestinal Fatty Acid Binding Protein experimental spectrum and best fit from SSNN-47.

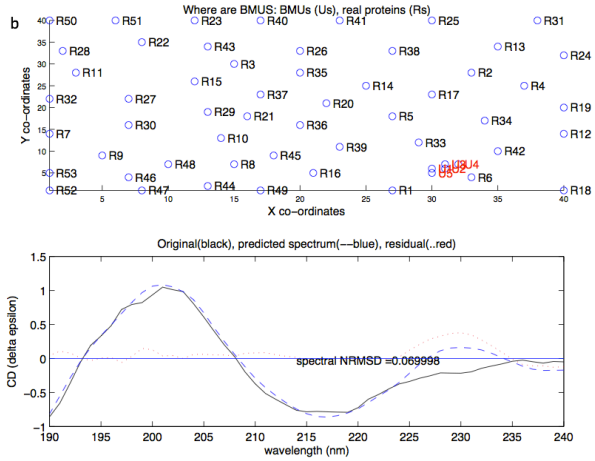
Table 2. Structural NRMSDs for SSNN-52 and CDSSTR-47 from the data in Table SI1 for different structural classes of protein. Here, except for 'Overall', there are only 3 structural classes α-helix, β-sheet and 'Other'. "Errors" are one standard deviation of the variation between proteins in the class.

Program	Overall 6-column	> 50% α-helix	30%–50% α-helix	>30% β-sheet	>50% Other
SELCON3-47	0.2±0.2	0.1±0.1	0.3±0.3	0.3±0.2	0.2±0.2
CDSSTR-47	0.3±0.2	0.2±0.2	0.3±0.2	0.2±0.2	0.3±0.2
SSNN-47	0.2±0.2	0.2±0.1	0.2±0.2	0.2±0.2	0.2±0.1
SSNN-52	0.1 ± 0.1	0.10±0.07	0.14±0.08	0.2±0.2	0.10±0.06

Some examples

To further illustrate SSNN and to give a new user some known examples to test their installation of the software, data sets for the biopharmaceutical product spectra of Figure 3 are included on the web site. The highly helical protein shows an exceedingly good fit with 83% helix and 0% sheet. The antibody spectral fit is not as good, but the key spectra features are reproduced with 5% helix and 38% sheet which is consistent with a slightly relaxed version of antibody crystal structures *e.g.* PDB 1IGT which is annotated by DSSP to be 6% helix and 47% sheet [18] and consistent with CDSSTR which suggests the spectrum is 3% helix and 35% sheet. The mixed structure asparaginase spectrum of Figure 3c has a small NRMSD and indicates 31% α-helix and 16% β-sheet, which compares with the crystal structure values of 31% and 23%.[19] The lower estimates of sheet content from CD and SSNN on solution data compared with x-ray diffraction on crystals are not surprising given the dynamic nature of proteins in solution.





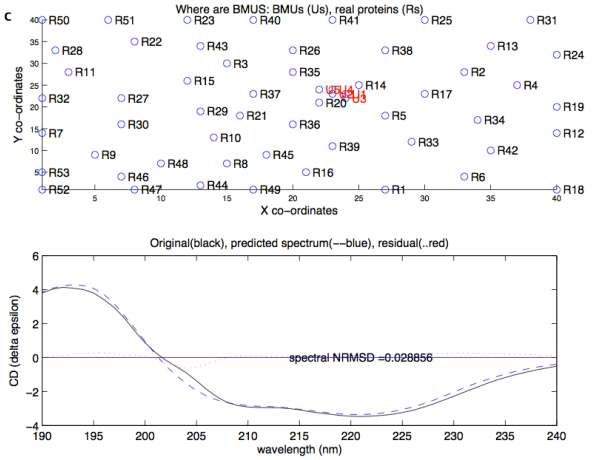


Figure 3. CD spectra and SSNN BMUs and model spectra for a range of biopharmaceutical product proteins. (a) A highly helical protein, SSNN structure vector (0.85, 0.07, 0.00, 0.00, 0.02, 0.05). (b) Antibody spectrum with structure vector (0.01, 0.04, 0.23, 0.15, 0.23, 0.34). (c) Asparaginase spectrum with structure vector (0.17, 0.14, 0.09, 0.07, 0.23, 0.31). Input data sets are available in SI.

Conclusions

We have made all parts of a neural network self organising map method of protein secondary structure determination from circular dichroism spectroscopy available a stand-alone program. SSNNGUI.app is a version that has been pre-trained with a currently available reference set. If a different reference set is desired, the first and second modules, SSNN1 and SSNN2 need to be run once for every new reference set used. These, and SSNN3 for structure estimation, are available in the second of two GUIs, named SSNN1_2.app. The output from the first two modules is internally used as input to SSNN3 which gives a secondary structure estimate for unknown proteins. All spectral data (reference set and test proteins) are formatted in our implementation as columns of 51 intensity points at 1 nm resolution from 240 nm to 190 nm in units of $\Delta\varepsilon$ (where the concentration is that of amino acids). Any other resolution or units can be sued as long as the reference and test spectra use the same. This means that a SOM secondary structure fitting

methodology is now available for use with new reference sets, *e.g.* for membrane proteins.

We have also shown that SSNN compares well with the statistical programs CDSSTR and SELCON3.[11] Although on average the methods can each be deemed to perform best for some secondary structures. For proteins known to have a high α-helix content, SELCON3 should be used to estimate structures. Both CDSSTR and SSNN are better at estimating β-sheet-rich proteins, and for intermediate and 'Other' proteins, SSNN is best. In practice by augmenting CDDATA.48 with 5 more reference spectra we have enhanced the performance of SSNN beyond that of the other methods in all but two cases. In practice, to obtain the most accurate picture of the structures of proteins it is advisable to use a few different secondary structure estimation methodologies, and trust them where a majority of them agree. If they agree then there can be confidence in the results and if they differ or the NRMSDs are high then care must be taken. Due to the simple metrics currently used by all fitting programs to assess goodness of spectral fit, we recommend complementing fitting programs with a visual inspection of the overlay of experimental data and model spectrum, particularly in the 215 nm region of the spectrum.

SSNN can be trained with pretty much any reference set for which structures are available. Thus SSNN could easily be adapted to a variety of data; examples could include CD data for membrane proteins, and infrared spectroscopy data.

Acknowledgements

We thank the Engineering and Physical Sciences Research Council for the funding for Vincent Hall and Anthony Nash through the MOAC Doctoral Training Centre (Grant number EP/F500378/1). Meropi Sklepari's help with the manuscript is gratefully acknowledged.

Notes and references

[a] *Molecular Organisation and Assembly in Cells Doctoral Training Centre, University of Warwick. Coventry, CV4 7AL, UK*
[b] *Department of Chemistry, University of Warwick. Coventry, CV4 7AL, UK. Fax: 44 24 76575795; Tel: 44 24 76574696; E-mail: a.rodger@warwick.ac.uk*
[c] *Anthony Nash is now at the Department of Chemistry, University College London, WC1H 0AJ*
[d] *Warwick Centre for Analytical Science, University of Warwick. Coventry, CV4 7AL, UK.*

† Electronic Supplementary Information (ESI) available: [details of any supplementary information available should be included here]. See DOI: 10.1039/b000000x/

1. N. Sreerama and R. W. Woody, *Analyt. Biochem.*, 1993, **209**, 32-44.
2. R. W. Woody, in *Circular dichroism principles and applications*, eds. K. Nakanishi, N. Berova and R. W. Woody, VCH, New York, 1994.
3. N. Sreerama and R. W. Woody, *Analyt. Biochem.*, 2000, **287**, 252–260.
4. L. Whitmore and B. A. Wallace, *Nuc. Acids Res.*, 2004, **32**, W668-673.
5. B. A. Wallace and R. Janes, eds., *Modern Techniques for Circular Dichroism Spectroscopy*, IOS Press, Amsterdam, 2009.
6. N. Berova, K. Nakanishi and R. W. Woody, eds., *Circular dichroism principles and applications*, 2nd edn., Wiley-VCH, New York, 2000.
7. R. W. Woody, *J. Am. Chem. Soc.*, 2009, **131**, 8234-8245.
8. S. W. Provencher, *Compter Phys. Comm.* , 1978, **27**, 229-242.
9. W. C. Johnson, *Proteins Struct. Funct. Genet.*, 1999, **35**, 307-312.
10. M. A. Andrade, P. Chacon, J. J. Merelo and F. Moran, *Prot. Eng.* , 1993, **6**, 383-390.
11. V. Hall, A. Nash, E. Hines and A. Rodger, *J. Comp. Chem.*, 2013.
12. A. Lobley, L. Whitmore and B. A. Wallace, *Bioinformatics* 2001, **18**, 211-212.
13. http://www.ogic.ca/projects/k2d3/, *accessed on 14th March 2012*.
14. C. Louis-Jeune, M. A. Andrade-Navarro and C. Perez-Iratxeta, *Proteins: Struc. Func. Bioinf.* , 2012, **80**, 374-381.
15. C. Perez-Iratxeta and M. A. Andrade-Navarro, *BMC Structural Biology* 2008, **8:25**.
16. P. Unneberg, J. J. Merelo, P. Chaco and F. M. n, *PROTEINS: Structure, Function, and Genetics*, 2001, **42**, 460-470.
17. B. A. Wallace, L. Whitmore and R. W. Janes, *Proteins-Structure Function and Bioinformatics*, 2006, **62**, 1-3.
18. L. J. Harris, S. B. Larson, K. W. Hasel and A. McPherson, *Biochemistry*, 1997, **36**, 1581-1597.
19. J. Lubkowski, M. Dauter, K. Aghaiypour, A. Wlodawer and Z. Dauter, *Acta Crystallogr.,Sect.D*, 2--3, **59**, 84.

Supplementary Information

SSNN, a method for neural network protein secondary structure fitting using circular dichroism data

VINCENT HALL,[a,b] ANTHONY NASH,[a,b,c] ALISON RODGER*[b,d]

[a] *Molecular Organisation and Assembly in Cells Doctoral Training Centre, University of Warwick. Coventry, CV4 7AL, UK*
[b] *Department of Chemistry, University of Warwick. Coventry, CV4 7AL, UK. Fax: 44 24 76575795; Tel: 44 24 76574696; E-mail: a.rodger@warwick.ac.uk*
[c] *Anthony Nash is now at the Department of Chemistry, University College London, WC1H 0AJ*
[d] *Warwick Centre for Analytical Science, University of Warwick. Coventry, CV4 7AL, UK.*

The supplementary information given below includes detailed instructions on how to install SSNN and also the detailed output from SSNNGUI.app and CDsstr for 48 proteins used in leave-one-out mode (SSNN1 and SSNN2 were run for the 48 sets of 47 spectra from CDDATA48).

Installing the MATLAB MCR and SSNNGUI

Download the appropriate version of SSNNGUI for your computer from http://www2.warwick.ac.uk/fac/sci/chemistry/research/arodger/arodgergroup/research_intro/instrumentation/ssnn/ (this may take some time). Unpack or unzip the SSNNGUI package, and use the MCR installer to install the MATLAB Compiler Runtime, so first unpack or unzip it. Unfortunately, different MCRs are required for different operating systems and the Mathworks MATLAB website only makes available the MCR going back a few versions. Installing the MATLAB Compiler Runtime on a PC should be straightforward if the correct one is chosen; the XP version of SSNN1_2 works fine on Windows 7 Enterprise. If it does not work, make sure you have run through the MCR installation process.

The situation is more complicated for a Macintosh. For the SSNN_Mac2013a release, one should install MATLAB R2013a MCR, as other MCRs are not compatible. To run this MCR installer launch <InstallForMacOSX.app> (the default will be to install it in the <Applications/MATLAB/MATLAB_Compiler_Runtime> folder). Follow instructions on screen until it asks to you to do something with the DYLD_LIBRARY_PATH and XAPPLRESDIR. It is *essential* that every character within the relevant <> is correctly typed from the following instructions. Towards the end of the MCR installation, the MCR installer displays messages about DYLD_LIBRARY_PATH and XAPPLRESDIR. The relevant changes can be made in the Terminal application (found within the Utilities folder of <Applications>). In Terminal, type <cd ../> press enter. Repeat typing <cd ../>, press enter then <ls> enter until the list includes <Application> in the list of files. Then type <cd Applications/MATLAB>. Then enter the following (see Figure S1): <export DYLD_LIBRARY_PATH=/Applications/MATLAB/MATLAB_Compiler_Runtime/v81/runtime/maci64:/Applications/MATLAB/MATLAB_Compiler_Runtime/v81/sys/os/maci64:/Applications/MATLAB/MATLAB_Compiler_Runtime/v81/bin/maci64:/System/Library/Frameworks/JavaVM.framework/JavaVM:/System/Library/Frameworks/JavaVM.framework/Libraries>. You should be fine if copying and pasting from the MCR installer window or this paper, but remember to include the <export DYLD_LIBRARY_PATH=> before the /Applications.

If an error message *e.g.* <not a valid identifier> appears, check every character in the above text Then type <export XAPPLRESDIR=/Applications/MATLAB/MATLAB_Compiler_Runtime/v81/X11/app-defaults>, also mostly copied from the MCR installer.

Using the command <env> will show you all of the environment variables, and these should be there now, as above. You can now quit Terminal.

Click <next> in the MCR installation window. Click <Finish>. The MCR is now installed.

Figure S1: The command to export the DYLD library path should look like this in Terminal if your computer is called Vince-Hall.

Using SSNNGUI with pre-trained SSNN1 and SSNN2

In the <SSNN_2013a_pkg> folder, launch the <SSNN_2013a.app> application. The best way to do this is using the shell. This is a file that is run in the command line, but does not require the user to input any command line commands. “Right-click” (or ctrl+tap touchpad in Mac) the <run_SSNN_2013a.sh> file, click <Open With Other…>, scroll to Utilities, then <Terminal.app>. The user might have to select Enable All Applications. Then click open. A Terminal window should open, and might display an error message. Do not worry about that, just wait a few second, and the SSNN application should launch.

Place the required test file in the correct format (see <example_3proteins.txt>) in the same folder as <SSNN_2013a.app>. The instructions in the Protocol of Table 1 should then be followed for a protein of known concentration.

To run SSNNGUI first locate the folder with the trained maps and the test spectra by entering the file path in the field marked with <Enter FULL_filepath_from_Users_or_root_no_slash> or click the <Browse> button to find it graphically (select any file within the correct folder). In the next field enter the name of the test file with no file extension (no .txt on the end). Enter the number of test spectra in the test file in the appropriate field. Click the “Run SSNN one dataset” button.

SSNNGUI has an option be run with automatic adjustment of concentrations of the test proteins, but that is not covered in this article. So here, ignore the <Run SSNN many concentrations> button, and the <Min concentration>, <Step concentration> and <Max concentration> fields.

The outputs from SSNN3 are the following files:

Spectral_NRMSD_vs_concentration_error_protein_1.pdf
best_concentrations.txt
all_NRMSDs.txt
all_NRMSDs_sorted.txt

Then in the folders with names like “53p40x40MAP_5_BMUs_28000ITER_0_06L0_60_Conc”:
Which_are_BMUs1_spectra_of_Winners.txt
Which_are_BMUs2_The_training_set.txt
Which_are_BMUs3_locations_of_Winners.txt
ssnn_residuals.txt
ssnn_real_spectra.txt
ssnn_predicted_spectra.txt
SSNN_parameters_extended.txt
RMSDspect.txt
plots_with_residuals

If SSNN is taking too long to run on a laptop computer, it can be put in standby and the run will continue when it is opened again. If there is a problem, open the log viewer (for example Console) to see more output, and any error messages. All of the windows can be closed after the plots of model against experimental CD spectra have been displayed. *Do not close any windows before SSNN has stopped running.* There should be one figure per protein spectrum if running the SSNN one data set, or wait for the NRMSD against concentration plots, one per protein spectrum if running the SSNN many concentrations. If you are running the program from the command line (this is the case if running it from the .sh file), then there will be a lot of output from the SSNN.

Running SSNN re-trainable, or "SSNN1_2"

Assuming that the appropriate MCR is installed (see Running SSNN3 above), one should be able to run SSNN1_2.app. When the user double-clicks on the SSNN application, it will take a few second to start up (Figure S2).

SSNN1

SSNN1 requires a reference set to be loaded in as a .txt file with the following format. Each protein CD spectrum and structure vector is a column of 57 entries (CD intensities for 240–290 nm at 1 nm spacing followed by 6 structure types summing to 1.00). The column vectors are placed side by side in the same file, in a tab or comma delimited format. For example, the reference set (CDDATA.48+5) used to train this version of SSNN has 53 proteins in an input file of size 57×53. That is CD intensities at 51 nm points, and 6 structure types, for 53 proteins. This input file is loaded using the SSNN graphical user interface (GUI, not "SSNNGUI"), designed for this purpose before SSNN1 is run. The structure information is not used in the training of SSNN1, only the spectral data are used at this stage. SSNN1 outputs a spectra map, which is then used by SSNN2 to give the corresponding structures map, and by SSNN3 to make models of the test protein spectra, and give structure estimations.

SSNN2

SSNN2 requires the same input file as SSNN1 as well as the SSNN1 output. The spectra in the input file are used to locate their representations on the spectra map, then the structures of each protein are added to the structure map at the same coordinates. SSNN2 outputs a structures map to correspond to the spectra map.

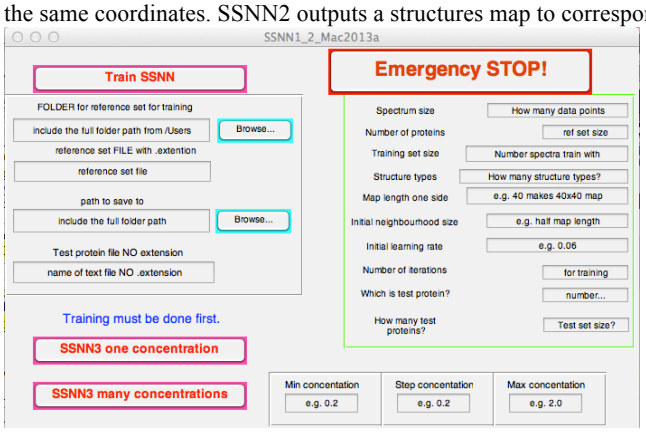


Figure S2: shows the SSNN1_2 application, which incorporates SSNN3.

SSNN1_2.app

The application SSNN1_2.app is available from http://www2.warwick.ac.uk/fac/sci/chemistry/research/arodger/arodgergroup/research_intro/instrumentation/ssnn/. It runs SSNN1, SSNN2, and SSNN3. The instructions below assume that the pretrained version SSNNGUI has been successfully run by the user. It is available in two Macintosh versions: MATLAB R2011a for Mac OS X 10.6.8 and MATLAB R2013a for OS X 10.8, as well as a MATLAB R2011b version for Windows XP, and a R2013a Windows 7 version. This application can be used to train SSNN with any data set of the correct format. We have provided CDDATA.48 and CDDATA.48+5 (see above). For the SSNN1_2_Mac2011a release (see below) the MATLAB R2011a MCR needs to be installed as described above.

To run the application, double click on the SSNN1_2*.app icon, then wait for the application to launch. Figure S2 shows the screen which controls the running of the application. SSNN1_2* means some file with a name that starts with "SSNN1_2".

i. In the top left there is an editable text field called <FOLDER for reference set for training>. For this field the user has two options: either to type in the full folder path from "/Users/" to the folder in which the reference set file is located or click on "Browse…" to navigate to it graphically. The reference set path and save path don't visibly update, but they are actually entered once the user has clicked on the "Browse…" button..

ii. The second field is for editable text (<reference set FILE with .extension>), this is the name of the

reference set file to train SSNN with. It should be named **with** the **extension**. It is made to take .txt files. One has to type in the reference set name, there is no option to select it graphically, this is for code reasons..

iii. The same options are available for the third field as for the first. This is called <path to save to>. This is a folder in which all output .txt files and figures will be saved, it can be the same as the reference path/folder. One can skip training if one changes the Weights and Structures folder date to today's, in which case they should not worry about entering anything into the second field.

iv. In the fourth field, <name of text file NO .extension>, the user should type the name of the test spectrum file **without** the .txt **extension**, for folder naming purposes in the code. This file should have number of rows matching "Spectrum size", and columns matching <How many test proteins>, and contain just spectra in the same range as the reference set file (in our case 51 rows of data from 240 – 190 nm in steps of 1 nm). One has to type in the test set name, there's no option to browse.

v. On the right side of the GUI, in the box with the thin green border, there are 10 fields. The first is "Spectrum size", this is where the user should tell SSNN1_2 how many wavelength points there are. (Data are in 1 nm steps in the reference sets provided.)

vi. The second field on the right is <Number of proteins> to tell SSNN1_2 how many columns there are in the reference set file loaded.

vii. Field 3 is <Training set size>. This will be smaller than the reference set size only if the user wants to train the SOM in leave-one-out-cross-validation; the SOM can be trained with one protein less than the full set size.

viii. <Structure types> is the number of structure types that have been assigned in the chosen reference set (6 in those supplied).

ix. <Map length one side> is so that SSNN can make a square map with the length or width of the number entered here.

x. < Initial neighbourhood size> is how much of the map should be modified in the first iteration of training to make it more similar to the best matching unit, BMU, at the centre of the neighbourhood and the protein spectrum that was chosen first. A neighbourhood is needed to cluster, otherwise dissimilar spectra would be near each other. The neighbourhood size will decrease with iterations to make the clustering more precise.

xi. <Initial learning rate> is how fast the BMUs will become more like the reference set. This decreases with iterations.

xii. < Number of iterations> this is the length of training of the SOM. We found that 28 000 iterations gave the best results (lowest structure estimation NRMSD) with our choice of map size, reference set size *etc.*.

xiii. <Which is test protein?> allows the user to exclude a protein (referenced by its number describing its position in the list) from the training set. If the user wants to train with the full reference set, then this field should be left empty, and <Training set size> should have the same number as <Number of proteins>. The 10^{th} field

xiv. < How many test proteins?> asks how many proteins will be in the test file.

xv. At the bottom of the GUI there are three fields that are to be used if the intensities of the test spectra need to be scaled, if for example the pathlength or concentration is likely to be in error. The fields are

xvi. <Min concentration>, < Step concentration>, and < Max concentration>. These are left empty if not required, they are for use when running "SSNN many concentration", and are not covered in this article.

To run the first two modules of SSNN1_2*.app, hit the <Train SSNN> button at the top left. Once the <Train SSNN> button has been clicked, there will be a pause before any output appears, then a window should pop up saying that 100 iterations have been completed, then further windows, counting up in 1000s from 1000. If you skip training, after modifying the name of the Weights and Structures folder to have today's date, then the figures will be much quicker to appear, see below.

SSNN3 part of SSNN1_2

Once the SOM has been trained by running SSNN1 and SSNN2 as outlined above, a <Weights_and_Structures…> folder will have been created. Then the <SSNN3 one concentration> button of SSNN1_2 can be clicked to produce the structure estimations for the test protein(s) as in the SSNNGUI version described above and in the main text. SSNN3 has finished running, when the final protein model against experimental CD spectrum plot pops up. This means one can close all the windows, there should be one for each spectrum in the test set.

For troubleshooting in Mac OS X 10.6.8, Console can be useful, as there will be some rather verbose output from SSNN1_2 appearing here. (This may not work for 10.8.x's Console, so try running it in the command line using the shell.) The Windows equivalent is Event Viewer. In Linux various applications are available including MultiTail, System Log Viewer, KSystemLog, Xlogmaster and swatch.

Table SI1: Structure vectors outputted from SSNN-47 and CDSSTR-47 for all proteins in CDDATA.48 from the CDPro web site tested in a leave-one-out methodology. SSNN was trained with 47 proteins 28,000 iterations, a map size of 40×40, initial neighbourhood of 20, BMUs = 5, $L_0 = 0.1$, $t_1 = 7{,}000$ iterations, and a $k_1 = 5\times10^{-6}$. CDSSTR-47 was run using the executable code available at http://lamar.colostate.edu/~sreeram/CDPro/main.html [1] using input files based on CDDATA.48 and SSDATA.48 but with one spectrum removed to be the test spectrum and structure each time. Average SELCON3 performance from [2] is also included.

Protein number	Protein name	Method	a-regular	a-distorted	b-regular	b-distorted	Turns	Other	Spectral NRMSD	Structural NRMSD	Fit minus real: a-regular	Fit minus real: a-distorted	Fit minus real: b-regular	Fit minus real: b-distorted	Fit minus real: turns	Fit minus real: other	CDSSTR NRMSD -
1	a-Bungarotoxin	Real	0.000	0.000	0.014	0.095	0.284	0.608	0.000								
		CDSSTR-47	-0.009	0.032	0.301	0.138	0.209	0.315	0.143	0.530	-0.009	0.032	0.287	0.043	-0.075	-0.293	0.432
		SSNN-52	0.015	0.032	0.086	0.102	0.239	0.526	0.362	0.098	0.015	0.032	0.072	0.007	-0.045	-0.082	
2	Alcohol Dehydrogenase	Real	0.139	0.115	0.139	0.096	0.214	0.297	0.000								
		CDSSTR-47	0.153	0.123	0.169	0.099	0.202	0.248	0.077	0.167	0.014	0.008	0.030	0.003	-0.012	-0.049	0.065
		SSNN-52	0.125	0.104	0.174	0.100	0.220	0.277	0.033	0.102	-0.014	-0.011	0.035	0.004	0.006	-0.020	
3	Adenylate Kinase	Real	0.340	0.206	0.077	0.052	0.012	0.313	0.000								
		CDSSTR-47	0.285	0.164	0.094	0.072	0.187	0.196	0.030	0.428	-0.055	-0.042	0.017	0.020	0.175	-0.117	0.203
		SSNN-52	0.277	0.180	0.074	0.062	0.116	0.291	0.021	0.225	-0.063	-0.026	-0.003	0.010	0.104	-0.022	
4	Azurin	Real	0.047	0.062	0.141	0.109	0.312	0.328	0.000								
		CDSSTR-47	0.118	0.104	0.213	0.121	0.195	0.241	0.115	0.545	0.071	0.042	0.072	0.012	-0.117	-0.087	0.437
		SSNN-52	0.073	0.076	0.170	0.104	0.271	0.307	0.073	0.108	0.026	0.014	0.029	-0.005	-0.041	-0.021	
5	b-lactoglobulin	Real	0.056	0.111	0.287	0.123	0.216	0.207	0.000								
		CDSSTR-47	0.100	0.090	0.155	0.102	0.211	0.337	0.048	0.319	0.044	-0.021	-0.132	-0.021	-0.005	0.130	0.193
		SSNN-52	0.074	0.110	0.245	0.119	0.218	0.233	0.046	0.126	0.018	-0.001	-0.042	-0.004	0.002	0.026	
6	Bence Jones Protein	Real	0.000	0.028	0.294	0.196	0.229	0.252	0.000								
		CDSSTR-47	-0.014	0.004	0.277	0.149	0.227	0.335	0.171	0.118	-0.014	-0.024	-0.017	-0.047	-0.002	0.083	0.032
		SSNN-52	0.003	0.035	0.267	0.172	0.221	0.303	0.263	0.086	0.003	0.007	-0.027	-0.024	-0.008	0.051	
7	Bovine Pancreatic Trypsi	Real	0.069	0.138	0.172	0.069	0.190	0.362	0.000								
		CDSSTR-47	0.122	0.096	0.060	0.044	0.117	0.561	0.114	0.198	0.053	-0.042	-0.112	-0.025	-0.073	0.199	0.114
		SSNN-52	0.074	0.112	0.147	0.071	0.173	0.422	0.137	0.084	0.005	-0.026	-0.025	0.002	-0.017	0.060	
8	Carbonic Anhydrase	Real	0.058	0.104	0.170	0.116	0.240	0.312	0.000								
		CDSSTR-47	0.023	0.034	0.096	0.054	0.108	0.679	0.060	0.255	-0.035	-0.070	-0.074	-0.062	-0.132	0.367	0.019
		SSNN-52	0.043	0.058	0.102	0.068	0.131	0.598	0.054	0.236	-0.015	-0.046	-0.068	-0.048	-0.109	0.286	
9	alpha chymotripsinoger	Real	0.053	0.082	0.210	0.110	0.210	0.335	0.000								
		CDSSTR-47	0.010	0.035	0.148	0.088	0.176	0.535	0.107	0.173	-0.043	-0.047	-0.062	-0.022	-0.034	0.200	0.003
		SSNN-52	0.050	0.073	0.147	0.078	0.153	0.498	0.068	0.171	-0.003	-0.009	-0.063	-0.032	-0.057	0.163	
10	a-Chymotrypsin	Real	0.069	0.045	0.208	0.106	0.200	0.371	0.000								
		CDSSTR-47	0.028	0.041	0.153	0.073	0.142	0.544	0.062	0.156	-0.041	-0.004	-0.055	-0.033	-0.058	0.173	0.077
		SSNN-52	0.050	0.043	0.175	0.097	0.198	0.437	0.107	0.080	-0.019	-0.002	-0.033	-0.009	-0.002	0.066	
11	Colicin A	Real	0.529	0.225	0.000	0.000	0.044	0.202	0.000								
		CDSSTR-47	0.524	0.280	0.032	0.018	0.051	0.083	0.023	0.110	-0.005	0.055	0.032	0.018	0.007	-0.119	0.061
		SSNN-52	0.499	0.198	0.007	0.012	0.085	0.198	0.024	0.049	-0.030	-0.027	0.007	0.012	0.041	-0.004	
12	Concanavalin A	Real	0.000	0.038	0.329	0.135	0.236	0.262	0.000								
		CDSSTR-47	0.077	0.076	0.187	0.120	0.205	0.336	0.234	0.290	0.077	0.038	-0.142	-0.015	-0.031	0.074	0.244
		SSNN-52	0.020	0.051	0.324	0.130	0.212	0.263	0.050	0.046	0.020	0.013	-0.005	-0.005	-0.024	0.001	
13	Carboxypepsidase A	Real	0.254	0.127	0.111	0.052	0.212	0.244	0.000								
		CDSSTR-47	0.122	0.108	0.221	0.123	0.223	0.204	0.075	0.680	-0.132	-0.019	0.110	0.071	0.011	-0.040	0.337
		SSNN-52	0.178	0.104	0.199	0.085	0.202	0.232	0.067	0.342	-0.076	-0.023	0.088	0.033	-0.010	-0.012	
14	Cytochrome C	Real	0.214	0.194	0.000	0.000	0.233	0.359	0.000								
		CDSSTR-47	0.204	0.161	0.100	0.076	0.171	0.290	0.053	0.305	-0.010	-0.033	0.100	0.076	-0.062	-0.069	0.145
		SSNN-52	0.198	0.154	0.063	0.056	0.205	0.324	0.054	0.161	-0.016	-0.040	0.063	0.056	-0.028	-0.035	
15	EcoR1 Endonuclease	Real	0.192	0.127	0.098	0.080	0.210	0.293	0.000								
		CDSSTR-47	0.209	0.147	0.104	0.078	0.169	0.291	0.049	0.094	0.017	0.020	0.006	-0.002	-0.041	-0.002	0.043
		SSNN-52	0.193	0.146	0.082	0.076	0.216	0.286	0.050	0.052	0.001	0.019	-0.016	-0.004	0.006	-0.007	
16	Elastase	Real	0.021	0.087	0.225	0.117	0.208	0.342	0.000								
		CDSSTR-47	-0.003	0.021	0.159	0.097	0.164	0.551	0.071	0.173	-0.024	-0.066	-0.066	-0.020	-0.044	0.209	0.001
		SSNN-52	0.019	0.057	0.152	0.083	0.155	0.534	0.092	0.172	-0.002	-0.030	-0.073	-0.034	-0.053	0.192	
17	Flavodoxin	Real	0.209	0.108	0.108	0.108	0.264	0.203	0.000								
		CDSSTR-47	0.136	0.110	0.145	0.092	0.210	0.309	0.040	0.273	-0.073	0.002	0.037	-0.016	-0.054	0.106	0.108
		SSNN-52	0.183	0.119	0.095	0.087	0.253	0.263	0.036	0.165	-0.026	0.011	-0.013	-0.021	-0.011	0.060	
18	g-Crystalin	Real	0.006	0.086	0.299	0.161	0.109	0.339	0.000								
		CDSSTR-47	-0.013	0.000	0.235	0.146	0.273	0.340	0.118	0.228	-0.019	-0.086	-0.064	-0.015	0.164	0.001	0.150
		SSNN-52	0.004	0.064	0.306	0.151	0.158	0.317	0.400	0.078	-0.002	-0.022	0.007	-0.010	0.049	-0.022	
19	Green Fluorescent Prote	Real	0.004	0.064	0.347	0.093	0.191	0.301	0.000								
		CDSSTR-47	0.074	0.086	0.219	0.131	0.241	0.249	0.158	0.393	0.070	0.022	-0.128	0.038	0.050	-0.052	0.348
		SSNN-52	0.023	0.058	0.334	0.110	0.192	0.283	0.093	0.045	0.019	-0.006	-0.013	0.017	0.001	-0.018	
20	Glyceraldehyde-3-phosp	Real	0.172	0.102	0.115	0.093	0.217	0.301	0.000								
		CDSSTR-47	0.178	0.127	0.094	0.071	0.172	0.360	0.029	0.119	0.006	0.025	-0.021	-0.022	-0.045	0.059	0.075
		SSNN-52	0.170	0.119	0.105	0.082	0.223	0.301	0.031	0.044	-0.002	0.017	-0.010	-0.011	0.006	0.000	
21	Glutathione Reductase	Real	0.188	0.142	0.140	0.096	0.172	0.262	0.000								
		CDSSTR-47	0.168	0.127	0.090	0.066	0.167	0.379	0.041	0.174	-0.020	-0.015	-0.050	-0.030	-0.005	0.117	0.029
		SSNN-52	0.148	0.123	0.126	0.096	0.214	0.293	0.034	0.144	-0.040	-0.019	-0.014	0.000	0.042	0.031	
22	Hemoglobin	Real	0.537	0.223	0.000	0.000	0.105	0.136	0.000								
		CDSSTR-47	0.527	0.296	0.039	0.011	0.062	0.062	0.014	0.095	-0.010	0.073	0.039	0.011	-0.043	-0.074	0.064
		SSNN-52	0.563	0.195	0.006	0.005	0.090	0.141	0.045	0.031	0.026	-0.028	0.006	0.005	-0.015	0.005	
23	Hemerythrin	Real	0.478	0.197	0.000	0.000	0.111	0.215	0.000								
		CDSSTR-47	0.416	0.220	0.070	0.060	0.128	0.104	0.020	0.183	-0.062	0.023	0.070	0.060	0.017	-0.111	0.144
		SSNN-52	0.478	0.192	0.024	0.018	0.105	0.184	0.040	0.039	0.000	-0.005	0.024	0.018	-0.006	-0.031	
24	Rat Intestinal Fatty Acid	Real	0.053	0.061	0.432	0.152	0.152	0.152	0.000								
		CDSSTR-47	0.113	0.103	0.140	0.105	0.263	0.279	0.065	0.808	0.060	0.042	-0.292	-0.047	0.111	0.127	0.051
		SSNN-52	0.151	0.095	0.229	0.101	0.205	0.219	0.076	0.757	0.098	0.034	-0.203	-0.051	0.053	0.067	

Protein number	Protein name	Method	a-regular	a-distorted	b-regular	b-distorted	Turns	Other	Spectral NRMSD	Structural NRMSD	Fit minus real: a-regular	Fit minus real: a-distorted	Fit minus real: b-regular	Fit minus real: b-distorted	Fit minus real: turns	Fit minus real: other	CDSSTR NRMSD –
25	Insulin	Real	0.294	0.235	0.020	0.040	0.050	0.361	0.000								
		CDSSTR-47	0.286	0.236	0.082	0.053	0.191	0.153	0.040	0.454	-0.008	0.001	0.062	0.013	0.141	-0.208	0.309
		SSNN-52	0.288	0.216	0.026	0.046	0.123	0.300	0.046	0.145	-0.006	-0.019	0.006	0.006	0.073	-0.061	
26	Lactate Dehydrogenase	Real	0.277	0.161	0.088	0.073	0.155	0.246	0.000								
		CDSSTR-47	0.293	0.176	0.077	0.062	0.165	0.228	0.036	0.060	0.016	0.015	-0.011	-0.011	0.010	-0.018	-0.028
		SSNN-52	0.290	0.179	0.083	0.067	0.117	0.264	0.025	0.088	0.013	0.018	-0.005	-0.006	-0.038	0.018	
27	Lysozyme	Real	0.202	0.217	0.016	0.047	0.298	0.221	0.000								
		CDSSTR-47	0.258	0.178	0.066	0.047	0.120	0.326	0.069	0.327	0.056	-0.039	0.050	0.000	-0.178	0.105	0.179
		SSNN-52	0.195	0.181	0.050	0.063	0.257	0.255	0.050	0.148	-0.007	-0.036	0.034	0.016	-0.041	0.034	
28	Myoglobin	Real	0.582	0.222	0.000	0.000	0.052	0.144	0.000								
		CDSSTR-47	0.587	0.306	0.021	0.003	0.035	0.049	0.010	0.091	0.005	0.084	0.021	0.003	-0.017	-0.095	0.031
		SSNN-52	0.659	0.172	0.003	0.002	0.051	0.114	0.031	0.060	0.077	-0.050	0.003	0.002	-0.001	-0.030	
29	Nuclease	Real	0.094	0.101	0.081	0.107	0.289	0.328	0.000								
		CDSSTR-47	0.182	0.141	0.083	0.058	0.166	0.375	0.047	0.220	0.088	0.040	0.002	-0.049	-0.123	0.047	0.089
		SSNN-52	0.133	0.121	0.096	0.096	0.249	0.305	0.037	0.131	0.039	0.020	0.015	-0.011	-0.040	-0.023	
30	Papain	Real	0.137	0.123	0.094	0.075	0.175	0.396	0.000								
		CDSSTR-47	0.040	0.032	0.143	0.108	0.222	0.449	0.079	0.158	-0.097	-0.091	0.049	0.033	0.047	0.053	0.104
		SSNN-52	0.122	0.112	0.111	0.084	0.195	0.376	0.094	0.055	-0.015	-0.011	0.017	0.009	0.020	-0.020	
31	Parvalbumin	Real	0.278	0.287	0.000	0.037	0.194	0.204	0.000								
		CDSSTR-47	0.315	0.215	0.067	0.041	0.130	0.232	0.057	0.188	0.037	-0.072	0.067	0.004	-0.064	0.028	0.069
		SSNN-52	0.268	0.240	0.018	0.038	0.182	0.254	0.078	0.119	-0.010	-0.047	0.018	0.001	-0.012	0.050	
32	Phosphoglycerate Kinas	Real	0.210	0.135	0.043	0.067	0.231	0.313	0.000								
		CDSSTR-47	0.432	0.217	0.032	0.051	0.110	0.153	0.040	0.317	0.222	0.082	-0.011	-0.016	-0.121	-0.160	0.186
		SSNN-52	0.259	0.159	0.049	0.058	0.197	0.279	0.056	0.131	0.049	0.024	0.006	-0.009	-0.034	-0.034	
33	Pepsinogen	Real	0.051	0.154	0.235	0.151	0.165	0.243	0.000								
		CDSSTR-47	0.019	0.041	0.251	0.126	0.230	0.332	0.074	0.214	-0.032	-0.113	0.016	-0.025	0.065	0.089	0.102
		SSNN-52	0.042	0.109	0.251	0.141	0.193	0.264	0.034	0.111	-0.009	-0.045	0.016	-0.010	0.028	0.021	
34	Prealbumin	Real	0.031	0.031	0.307	0.142	0.165	0.323	0.000								
		CDSSTR-47	0.013	0.057	0.253	0.116	0.241	0.314	0.075	0.139	-0.018	0.026	-0.054	-0.026	0.076	-0.009	0.073
		SSNN-52	0.036	0.055	0.313	0.136	0.172	0.288	0.037	0.066	0.005	0.024	0.006	-0.006	0.007	-0.035	
35	Rhodanase	Real	0.150	0.147	0.041	0.068	0.235	0.359	0.000								
		CDSSTR-47	0.218	0.148	0.118	0.080	0.183	0.247	0.018	0.394	0.068	0.001	0.077	0.012	-0.052	-0.112	0.298
		SSNN-52	0.188	0.151	0.062	0.067	0.207	0.325	0.030	0.096	0.038	0.004	0.021	-0.001	-0.028	-0.034	
36	Ribonuclease A	Real	0.113	0.097	0.218	0.113	0.218	0.242	0.000								
		CDSSTR-47	0.116	0.092	0.144	0.096	0.220	0.332	0.069	0.201	0.003	-0.005	-0.074	-0.017	0.002	0.090	0.088
		SSNN-52	0.104	0.093	0.186	0.108	0.228	0.280	0.087	0.112	-0.009	-0.004	-0.032	-0.005	0.010	0.038	
37	Substilin BPN	Real	0.171	0.131	0.098	0.080	0.225	0.295	0.000								
		CDSSTR-47	0.119	0.089	0.163	0.105	0.196	0.328	0.056	0.181	-0.052	-0.042	0.065	0.025	-0.029	0.033	0.151
		SSNN-52	0.166	0.126	0.094	0.078	0.228	0.309	0.056	0.030	-0.005	-0.006	-0.004	-0.002	0.003	0.014	
38	Substilin novo	Real	0.113	0.102	0.065	0.073	0.295	0.353	0.000								
		CDSSTR-47	0.211	0.136	0.140	0.097	0.190	0.230	0.046	0.637	0.098	0.034	0.075	0.024	-0.105	-0.123	0.452
		SSNN-52	0.178	0.155	0.063	0.063	0.223	0.318	0.058	0.185	0.065	0.053	-0.002	-0.010	-0.072	-0.035	
39	Superoxide Dismutase	Real	0.000	0.018	0.248	0.119	0.298	0.316	0.000								
		CDSSTR-47	0.031	0.047	0.253	0.138	0.214	0.312	0.131	0.140	0.031	0.029	0.005	0.019	-0.084	-0.004	-0.036
		SSNN-52	0.064	0.066	0.204	0.109	0.236	0.321	0.088	0.176	0.064	0.048	-0.044	-0.010	-0.062	0.005	
40	T4 Lysozyme	Real	0.421	0.244	0.049	0.037	0.116	0.134	0.000								
		CDSSTR-47	0.269	0.165	0.123	0.079	0.187	0.175	0.026	0.447	-0.152	-0.079	0.074	0.042	0.071	0.041	0.348
		SSNN-52	0.384	0.216	0.050	0.037	0.109	0.204	0.036	0.099	-0.037	-0.028	0.001	0.000	-0.007	0.070	
41	Thermolysin	Real	0.282	0.133	0.070	0.095	0.215	0.206	0.000								
		CDSSTR-47	0.258	0.156	0.107	0.065	0.148	0.270	0.028	0.218	-0.024	0.023	0.037	-0.030	-0.067	0.064	0.029
		SSNN-52	0.247	0.166	0.062	0.065	0.178	0.282	0.031	0.189	-0.035	0.033	-0.008	-0.030	-0.037	0.076	
42	Tumor Necrosis Factor	Real	0.000	0.019	0.293	0.140	0.219	0.329	0.000								
		CDSSTR-47	-0.003	0.019	0.226	0.139	0.221	0.386	0.180	0.092	-0.003	0.000	-0.067	-0.001	0.002	0.057	0.053
		SSNN-52	0.008	0.038	0.286	0.147	0.208	0.314	0.145	0.040	0.008	0.019	-0.007	0.007	-0.011	-0.015	
43	Triose Phosphate Isome	Real	0.236	0.210	0.090	0.064	0.124	0.276	0.000								
		CDSSTR-47	0.349	0.180	0.075	0.061	0.165	0.175	0.023	0.228	0.113	-0.030	-0.015	-0.003	0.041	-0.101	0.114
		SSNN-52	0.284	0.198	0.072	0.057	0.142	0.247	0.018	0.114	0.048	-0.012	-0.018	-0.007	0.018	-0.029	
44	Apo-cytochrome C (5°C	Real	0.020	0.020	0.020	0.020	0.020	0.900	0.000								
		CDSSTR-47	0.006	0.031	0.082	0.052	0.096	0.725	0.068	0.116	-0.014	0.011	0.062	0.032	0.076	-0.175	0.106
		SSNN-52	0.020	0.021	0.027	0.023	0.027	0.882	0.017	0.010	0.000	0.001	0.007	0.003	0.007	-0.018	
45	Apo-cytochrome C (90°C	Real	0.020	0.020	0.020	0.020	0.020	0.900	0.000								
		CDSSTR-47	0.038	0.034	0.075	0.060	0.113	0.673	0.045	0.163	0.018	0.014	0.055	0.040	0.093	-0.227	0.014
		SSNN-52	0.030	0.044	0.086	0.055	0.097	0.687	0.068	0.149	0.010	0.024	0.066	0.035	0.077	-0.213	
46	Ribonuclease (20°C) den	Real	0.020	0.020	0.020	0.020	0.020	0.900	0.000								
		CDSSTR-47	0.004	0.036	0.082	0.048	0.101	0.726	0.066	0.116	-0.016	0.016	0.062	0.028	0.081	-0.174	0.108
		SSNN-52	0.020	0.021	0.024	0.022	0.027	0.886	0.039	0.008	0.000	0.001	0.004	0.002	0.007	-0.014	
47	Staphyllococcal Nucleas	Real	0.020	0.020	0.020	0.020	0.020	0.900	0.000								
		CDSSTR-47	-0.004	0.026	0.054	0.038	0.066	0.809	0.048	0.056	-0.024	0.006	0.034	0.018	0.046	-0.091	0.038
		SSNN-52	0.021	0.023	0.032	0.025	0.032	0.867	0.052	0.018	0.001	0.003	0.012	0.005	0.012	-0.033	
48	Staphyllococcal Nucleas	Real	0.020	0.020	0.020	0.020	0.020	0.900	0.000								
		CDSSTR-47	0.013	0.030	0.091	0.060	0.115	0.688	0.053	0.149	-0.007	0.010	0.071	0.040	0.095	-0.212	0.060
		SSNN-52	0.028	0.039	0.060	0.043	0.072	0.758	0.059	0.089	0.008	0.019	0.040	0.023	0.052	-0.142	
	Sum of absolute values	SELCON3-47							4.480	10.560	1.838	1.651	2.784	1.316	2.980	4.488	
	Sum of absolute values	CDSSTR-47							3.301	12.119	2.166	1.714	3.131	1.240	3.054	5.174	
	Sum of absolute values	SSNN-47							4.380	10.800	2.230	1.910	2.640	1.310	2.790	4.670	6.309
	Sum of absolute values	SSNN-52							3.525	5.809	1.097	1.017	1.332	0.636	1.433	2.341	

1. V. Sreerama N., S.Y.U., Woody RW., *Protein Science*, 1999, **8**, 370-380.
2. V. Hall, Nash, A., Hines, E., Rodger, A., *Journal of Computational Chemistry*, 2013, **34**, 2774-2786.

Self organising map pattern recognition for robotic prosthetic control: HASSANN

Vincent Hall[1-2], Max Ortiz-Catalan[3-4]

1. MOAC, University of Warwick, Coventry, CV4 7AL, UK.
2. Department of Chemistry, University of Warwick, Coventry, CV4 7AL, UK.
3. Department of Signals and Systems, Biomedical Engineering Division, Chalmers University of Technology, Gothenburg, Sweden
4. Centre of Orthopaedic Osseointegration, Department of Orthopaedics, Sahlgrenska University Hospital, Gothenburg, Sweden

Abstract

Tens of thousands of people have limbs amputated each year, causing great loss of ability in people's lives. [1] Pattern recognition of myoelectric signals has the potential to restore the functionality of a missing limb by providing an intuitive control of an artificial one. Here we report on **HASSANN:** Hand Activation Signals SOM Artificial Neural Network, a self-organising map (SOM) methodology for performing the myoelectric pattern recognition stage of BioPatRec. BioPatRec is a software platform for processing myoelectric signals to control prosthetic arms and hands. [2] HASSANN has been evaluated in a classification task of 11 individual hand and wrist motions. In offline testing of how data patterns are recognised, it was found that all movement classes could be recognised from the myoelectric signals. The HASSANN methodology is ready to be tested in real-time control of prosthetics. The accuracy is high in offline testing, 0.90 ± 0.08, which would likely be accurate enough for use by a patient. However, algorithm accuracy usually reduces significantly when performing real-time testing. So the real-time tests will show if this software can be used in its current form.

Introduction

Amputations and prosthetics

Every year there are about 2,000 arm amputations, and about 65,000 leg amputations at the Symes level (the ankle above both malleoli or bumps [3]) and higher in the USA, according to Brenner et *al*. 2008. [1] The number of amputations in the UK is around 6000 a year. [4] According to Le Blanc et *al*. [5] , in 2008 there were an estimated 10 million amputees living in the world, 30 % of these being arm amputees, so 3 million people, 2.4 of which were in developing countries. Of the arm amputations, 59 % were below the elbow, 28 % above the elbow, 8 % were to shoulder level, 5 % were hand and wrist.

With such a large number of people missing limbs, along with those born with limb defects that cause loss of function, there is great need to provide replacement limbs the enable much of the use of natural limbs as possible.
Currently very little in the way of robotic arms is available commercially, and most of those that are available have a simple open hand–close hand movement. In fact, today there is no commercially available arm/hand that uses pattern recognition-based control.

The prosthetics community seems to agree that there are five basic types of grasp that the human hand performs daily:

1. The pincher grasp (with thumb, index and middle fingers together)
2. Key grasp (thumb resting on the side of the index finger)
3. Hook grasp (used to carry things e.g. books, suitcases)
4. Spherical grasp (for holding a ball)
5. Cylindrical grasp (for holding a cylindrical object) [6]

The most advanced prosthetic hand control technology that is available clinically depends on myoelectric signal (MES) control. [7] The free dictionary defines myoelectric as being "of or pertaining to electrical impulses generated by muscles of the body, which may be amplified and used esp. to control artificial limbs." [3]

BioPatRec

In this paper, we report on a software pattern recognition methodology developed for control of robotic prostheses of the upper limb that use myoelectric signals, it is labelled HASSANN, and will be part of the BioPatRec software platform. Any artificial electrical control of movements needs to be able to recognise patterns to determine which ways the person wishes to move, and should provide as natural limb movement as possible. BioPatRec with HASSANN is one such methodology.

BioPatRec can be found here: https://code.google.com/p/biopatrec/. It is a platform used for collaboration on robotic prosthetic limb control. At this point the limiting factor in producing replacement limbs that fully compensate for natural limbs, with the complete range of motions and abilities is the control software. BioPatRec is available to provide this control. The platform is meant for testing control software; so there is no one prosthetic being studied by the BioPatRec team. The aim is to use it for a wide variety of limb designs. The end goal is to have robotic limb control algorithms robust enough for clinical implementation, which will be put into microcontrollers that can run MATLAB.

The aim of this work has been to provide a new development to the pattern recognition stage of BioPatRec called HASSANN, Hand Activation Signals SOM Artificial Neural Network. HASSANN has been designed to be used for the 4th stage of BioPatRec: pattern recognition, which is the core stage (see the stages below). HASSANN is not yet available on the BioPatRec website.

HASSANN was derived from software for performing pattern recognition on circular dichroism (CD) spectroscopic data to estimate protein secondary structures. [8-10] The Secondary Structure Neural Network, SSNN, software that HASSANN is based on can be found here: http://www2.warwick.ac.uk/fac/sci/chemistry/research/arodger/arodgergroup/research_intro/instrumentation/ssnn/. SSNN has three modules: one for clustering the spectral data, one for mapping the *structures* of the proteins to their CD spectra, and one to accept query or test spectra with unknown structures in order to model them, and provide estimations of those unknown structures.

As SSNN is a SOM methodology, it can be applied to essentially any data set. A SOM is a Machine Learning clustering methodology that makes high-dimensional data easier to view, it also preforms pattern recognition to condense that high-dimensional data into

a small amount of easy-to-understand, low-dimensional knowledge for the user. We have adapted the SSNN code (and renamed it HASSANN) to be a software package that can be used to interpret pre-processed electrical signals from muscles to control robotic limbs. We have used the MES recorded by the BioPatRec team to train HASSANN and to test it. These data are available in the Data Repository folder in the BioPatRec_ETT.zip download on the website: reference [11]. This repository enables the comparison of different pattern recognition methodologies. [12]

Details of HASSANN and BioPatRec

HASSANN was trained with the data from the BioPatRec data repository once earlier stages of BioPatRec had pre-processed it.
There are 5 stages that the BioPatRec data go through:

1. Signal recording
2. Signal processing
3. Feature selection and extraction
4. Pattern recognition
5. Real-time limb control

These five stages are how the software platform runs, but let us explain the advantages of BioPatRec and HASSANN. While literature agrees that at least 5 movements are used by the hand, BioPatRec works with 11 pre-set movements (10 active and 1 rest), see the list below:

1. open hand
2. close hand
3. flex hand (move hand from wrist in direction of palm)
4. extend hand (move hand from wrist in direction of back of hand)
5. pronation (place palm down)
6. supination (place palm up)
7. side grip or hook grasp (e.g. for opening a fridge door/carrying suitcases)
8. fine grip (similar to pincher)
9. agree (thumb up, fingers bent in)
10. pointer (index finger pointing)
11. rest (resting position of hand)

So more movements are available than the literature says are required, and there are many advantages to using myoelectric data, as BioPatRec does. The method of gathering the neuromuscular data is non-invasive, involving just the placement of electrodes on the skin. The electrode detects the passing of the action potential through the muscle fibres. Each electrode stores the value of the sum of all action potentials once they have travelled through the muscle fibres to the surface of the skin. [6] A key element of the advantage of using myoelectric data is the autonomous nature of control of limbs in a manner similar to natural movement. [13] However, when using certain prosthetic limbs for the first time, the patient needs to be trained, as the muscles that emit the control signals are not necessarily exactly the same as those in the recordings in the reference set, or those signals used in the normal function of a natural, biological arm. Different people might have different motor unit structures (see “Electrical activity of natural muscles”). As such, use of the prosthetic will require lots of concentration. [7,14] HASSANN, and BioPatRec seek to minimise this effort by intelligently interpreting the signals for quicker recognition.

Some earlier prosthetics might require the hand to return to the starting or rest position before moving to the next position. A limb controlled by BioPatRec can switch from one movement to the next without having to rest first. There are videos of prosthetic hands controlled by BioPatRec being tested and demonstrated on the YouTube channel NCALOI. [15]

Other methodologies

Our software uses pre-processing, and it is generally good practice, due to the huge size and complexity of the MES. This is because dimensionality reduction needs to be performed on the data.
Pattern recognition algorithm accuracy in defining the correct classes (or movements) hinges mostly on the representation of the continuous time waveforms as feature vectors.
Therefore, feature vectors must be selected to reduce to a minimum the error of controlling the limbs. A feature set that distinguishes between movements as clearly as possible needs to be selected. [16]

Various methods of pre-processing have been employed, Chu *et al*. in reference [17] used PCA and SOM for the non-linear feature projection, followed by an MLP for the classification stage. An MLP is a multilayer perceptron, which is a type of artificial neural network that is trained by making guesses, and having those corrected against the known answers. This is called supervised learning.
The use of PCA simplifies the structure of the classifier, and saves computational time. This is followed by SOM non-linear mapping, because they find it makes it easier to separate the different classes, compared with PCA alone. Then the final stage is the MLP for classification.

This is a wise architecture for machine learning (ML), as it has been found that applying successive rounds of ML techniques to a data set produces the best results. For example that used by Schmidhuber [18], where unsupervised–learning methodologies are applied in a hierarchical manner. In unsupervised learning, a methodology is used to study data where the correct classifications are not known (data mining). The aim here is to take unlabelled data, find patterns in it, and come up with new correlations or relationships. This is opposed to supervised learning (which uses labelled data), where the methodology is corrected each step, and the aim is generalisation to unknown data.

The focus of the work by Schmidhuber is only on the errors, and in correcting them. The number of errors reduces each layer, with the output from layer n being input for layer n+1. Therefore, the higher-level predictors have less difficulty predicting the input to the next layer than do the lower level predictors.

A hierarchy of statistical and or machine learning techniques may produce much better classification, but would also take a lot more computational time, and computational time must be kept low for things used in real-time. When a receiver of a prosthetic limb is using it, they need the limb to respond quickly, low latency. A response time of more than about 300 ms would result in unacceptably low speeds of movement. [16]

Electrical activity of natural muscles

In the body, myoelectric signals, or electromyographic (EMG) signals, are generated by skeletal muscles when they are electrically or neurologically stimulated. These muscles are attached to the tendons and bones, and are under conscious control, as they are part of the somatic nervous system. They also come under subconscious control for regulatory purposes. However, the signals that cause skeletal muscle activation always originate with nerve impulses. [6]

Change in force applied by muscles is made possible by adding together responses to a series of stimuli and by employing more motor units or SMUs. An SMU, or single motor unit, is a collection of muscle fibres and the motor neuron that innervates the muscle fibres. The SMUs are naturally arranged to contain varying numbers of muscle fibres: those for gentle, dexterous motions have only 3–10 fibres, while others used for large, forceful motions contain several hundred muscle fibres.

An action potential in an SMU can cause a twitch, but continued muscle contraction needs many firings. When a ramping up of muscle power is needed, the smallest muscle units will be employed first, then larger and larger units, until the full force of all units is applied. [6,19] The resting potential of skeletal muscles is about –70 mV to –90 mV, which is similar to that of a neuron.

The instantaneous myoelectric signal obtained from the electrodes contains no information, as it is stochastic, so a contraction needs to be recorded over some time. This time is usually in the 100s of ms.[16] The variance of the MES depends on the level of contraction of the muscles. Harder contractions produce more variance. [20] Due to differences in structure and size of different people's muscles, there are differences between people in the myoelectric signals received from them, and therefore the features extracted from that data. Of course, amputation and congenital defects can greatly change the shape, size and structure of muscles. Changes in weight and/or the positioning of the electrodes might also modify the MES recorded. Therefore the pattern recognition, or classifier, methodology must be able to allow for these differences, and remain efficient in its functioning. Certain movements produce rather deterministic patterns, while other contraction types are nearly random in nature. [16]

Methods

BioPatRec has previously tested various pattern recognition algorithms: Regulatory Feedback Networks (RFN), Linear Discriminant Analysis (LDA), multilayer perceptron (MLP). [12] Other classification methodologies used to perform pattern recognition on MES include hidden Markov models (HMM), and Gaussian mixture models (GMMs), dynamic artificial neural networks, genetic algorithms, fuzzy logic classifiers, PCA and self-organising maps. [13,17]

Glossary

BMU – best matching unit, the map resident spectra that go into making the model of the spectrum.
DoF – degrees of freedom
EMG –electromyogram
GMM – Gaussian Mixture Model
GUI – Graphical User Interface
HMM – hidden Markov models

LDA – Linear Discriminant Analysis
MES – myoelectric signals
MLP – MultiLayer Perceptron
RFN – Regulatory Feedback Networks
PCA – Principal Component Analysis
SMU – single motor unit, the fibres and neurons that power muscles
SOM – Self-Organising Map, also called Kohonen Map, SOFM, where "F" stands for feature.

Data acquisition and signal processing for BioPatRec

The BioPatRec team recorded the myoelectric signals by using 4 and 8 disposable electrodes, for individual and simultaneous motions respectively. These were equally spaced around the most proximal third of the forearm, one proximal and one distal in each pair. The electrodes are disposable, and bipolar. They are made of silver and silver chloride (Ag/AgCl), and their diameter is 1 cm. The positive terminal of the bioelectric amplifier was consistently connected to the most proximal electrode (proximal to the body).

The myoelectric signals were segmented into time windows of 200 ms, from which 4 signals features were extracted (mean absolute value, wave length, zero crossing, and slope sign changes). If the mean absolute values of *entire* time course recordings were used, they would lose too much of the features, and end up with flat signals containing no information. Most signal means would look too similar, which would make classification nigh impossible. Further details in the data acquisition, signal processing and data repository can be found in reference [21].

The features from all channels for a specific time window form a feature vector characterizing a particular movement. The feature vectors are fed into the pattern recognition algorithm rather than the raw myoelectric signal itself. This is because trying to use the raw data for pattern recognition would produce classification of movements with very poor success rates, which would cause the prosthetics to be unusable, remember that the response time should be under 300 ms. [16] The recordings from one patient are 36,000 by 4 by 10 data elements, and there are 20 patients.

The data recorded can also be cropped so that areas with no activity are cut out. This is measured by the cTp, or contraction time percentage, which is usually about 70. So, 15 % of the time–series is cut from the beginning of the recording, and another 15 % from the end of the recording. [12]
The pattern recognition algorithms in BioPatRec are given 27 signal features in the time, and in the frequency domains. [12]

Data sets

For presentation to the pattern recognition methodology, the data were divided into 12 training sets, 6 validation sets and 6 test sets when using each patient's data set from the repository named "10mov4chForearmUntargeted". This repository is for 10 movements plus rest, with 4 electrodes producing 4 channels placed on the forearm. Untargeted refers to the placing of the electrodes on the arm, see reference [12]. Four electrodes were enough because the use of just 4 electrodes has been tested and found to be sufficient

for 10 movements. The "rest" is a standard signal that can be added for any movement. The signals were digitised at 2 kHz with a resolution of 14-bits.
The data was recorded while the subject performed the contraction for 3 seconds, then rested for 3 seconds, then repeated this 3 times. [12,21-23]

Summary of SOM architecture used by HASSANN

Once the datasets were compiled, and the software was ready to be adapted to the MES data.
As a SOM, HASSANN performs its pattern recognition by clustering the data into groups where the signals have similar features. It then looks up the signals again, and assigns a value of 1 to the training set signals only (not interpolations between MES). These are known to belong to certain categories (movements).
With the BioPatRec myoelectric data the signals each correspond to patients attempting to perform 1 of 11 movements (10 hand and wrist movements plus the resting position). HASSANN tags each signal with a vector of 11 numbers summing to 1.00. For signals corresponding to the n^{th} movement there will be a 1 at the n^{th} position, and a zero in each of the other 10 positions.
In the final stage, when HASSANN decides which movement is desired, judging by the signal received, it will give a vector of 11 numbers that sum to 1.00. The largest number will correspond to the movement it has decided is desired. In this sense, the output is similar to that of a fuzzy logic methodology. This decision will be given to the next stage of BioPatRec to deal with: the Real time control software.

More explicit HASSANN methodology

The SOM was designed by Teuvo Kohonen in 1982, and is known as the Kohonen Map, or the Self-Organising Feature Map (SOFM). [25]

For a full description of the HASSANN methodology please refer to earlier papers on this by Hall and Rodger *et al.* [8,9,24] In these references, the methodology of SSNN, Secondary Structure Neural Network, is discussed. However, here is a more detailed account of how HASSANN operates.

Stage 1: clustering signal vectors

HASSANN is a methodology for clustering data sets using a 40 by 40 node square grid. Here, each myoelectric time-course signal is represented by a vector, which is placed on a map (in a node) to cluster it with all other signals. This is done after the BioPatRec pre-processing stages, so only the features of the MES are clustered.
To begin with, each map node has a vector of pseudorandom numbers populating it. Next, a randomly selected pre-processed signal of 52 numbers is shown to the map, and the most similar vector of random numbers is found; this is called the BMU, or best matching unit.

Using a learning rule, every element of the BMU vector on the map is made more similar to the signal vector selected. A neighbourhood of the BMU is defined based on the initial neighbourhood radius, and the neighbouring nodes are made more similar using the same learning rule, but weighted so that nodes closer to the BMU will learn faster than those at the periphery of the neighbourhood.
This is repeated for a set number of iterations. This number is usually thousands, but it must be much larger enough that each of the signal vectors in the training set is represented, and there is time to learn. The learning rate each iteration is about 0.08 to 0.1, so several iterations are required for each sample vector. This ends the longest stage of training. By this point, the map or SOM should have randomly selected each pre-processed signal vector many times, and clustered all signals into groups. There are no distinct boundaries, like fuzzy logic where each sample will have several memberships, all summing to 1.00.

There are many more virtual myoelectric signal nodes than experimental signal nodes, as the map interpolates between signals to allow for slight changes in each signal that might represent the pre-set movements with noise or slight changes. This also allows for missing data.

Stage 2: tagging map vectors with movements

The trained SOM now needs to be told which signals correspond to which of the 11 pre-set movements:

To do this HASSANN searches its map for the BMUs of the training set vectors (neglecting the virtual vectors for this first part) and assigns to each a vector of 11 numbers summing to 1.00, with a '1' in the n^{th} position and '0's in the other 10 possible positions for the n^{th} movement. These movement vectors are in a second map with coordinates that correspond to the first map. The 11-element movement vectors come from the training set.
Next, HASSANN looks for each virtual signal node on the first map, makes a weighted sum of the movement vectors of the 5 nearest experimental signals from the second map, and assigns this sum as the movement vector for the virtual signal node on the second map. At the end of stage 2 all signal vector nodes (experimental and virtual) have movement vectors associated with them in a second map with the same coordinates as the first.

Stage 3: deciding which movement is desired

Once HASSANN's SOM is fully trained, it can be used to decide which movement is needed by the patient with the prosthetic arm, depending on the myoelectric signals it receives from the patient via the electronics. The real-time control uses the SOM to interpret the signals input to it. HASSANN finds the b BMUs of the real-time* data, and makes a weighted sum of the movement vectors and outputs that as the decision of

* This is done in an offline test using just the recorded data, and then in a real-time test with real-time/live limb control.

which movement to make. (Here “b” is the number of BMUs used in the calculation.) The same is done to make a model MES spectrum, as can be seen in Figure 1(d).
The most important text output is the decision vector, which is in the format of the input movement data: The elements of this vector of 11 numbers are the 11 movements: Ideally a 1.00 for the element corresponding to the decided upon movement, and zeros for movements HASSANN decides are not needed at this time. More likely output would be vector of elements with numbers between 0 and 1, where a certain element has the highest number, and therefore wins, thus telling the BioPatRec system to tell the robotic arm to move using that movement. The output movement vector *always* sums to 1.00.

What HASSANN outputs look like

The model of the test spectrum is plotted along with the test spectrum itself, and a residual. There is also an NRMSD value on the plot, see Figure 1(d). The NRMSD is the normalised root mean squared deviation, or the RMSD divided by the range of the model data, see equation 1, below.

$$NRMSD = \frac{\sqrt{\left(\frac{\sum_i \left(x_{i,experiment} - x_{i,model} \right)^2}{N} \right)}}{M - m} \quad (1)$$

where x_i counts the elements of the MES spectrum vector, experimental or model, N is the number of elements in the spectrum, M is the maximum number, and m the minimum number in the model data.

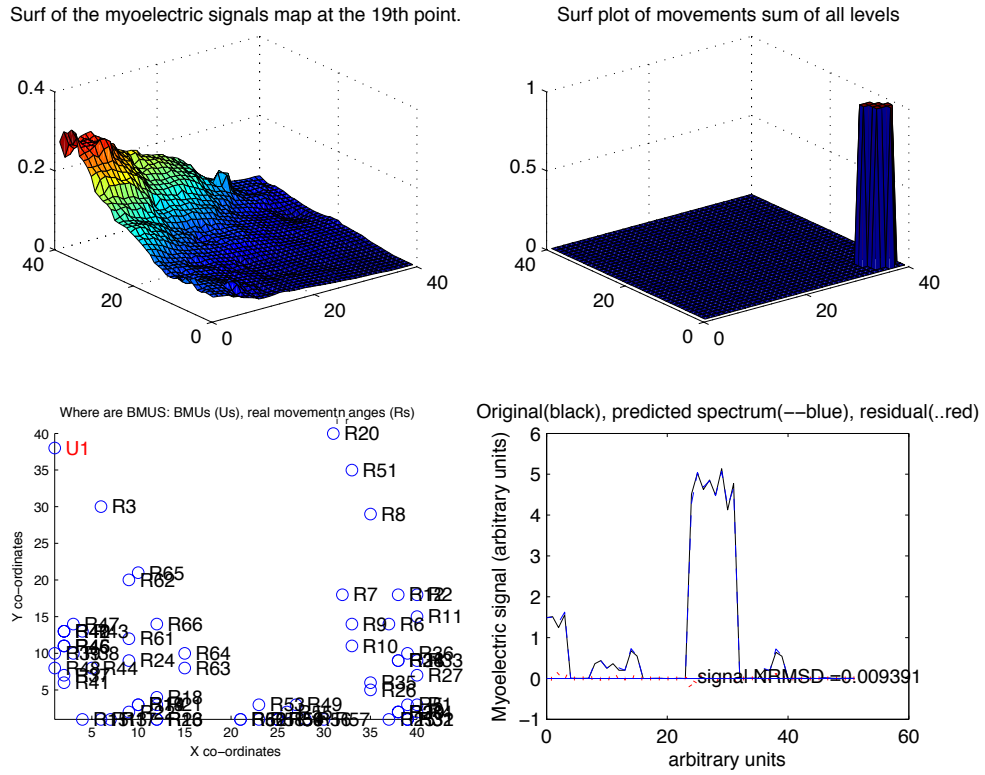


Figure 1: What a good result from the MES clustering should look like. Figure a) is the 19th level of the clustered MES spectra map, b) is the movements map (this shows the movements are all clustered in the right corner), c) shows the locations of the *training* set MES BMUs marked with R#, and the MES spectrum that went into making the *model* MES spectrum for this particular test is marked with a U# in red. d) shows the model MES spectrum in blue (dashed line), the experimental spectrum in black (solid line), the residual in faint red (dotted line at the bottom), and the NRMSD.

Characteristics of HASSANN training

The way the map clusters in Stage 1 depends on the learning equation:

$$learning = L_0 * exp^{-k_1 * t} \tag{2}$$

where L_0 is the initial learning rate, approximately the rate of the first iteration. L_0 was set to 0.1. k_1 is a small value that depends on L_0 and the number of iterations: from t to T, the final iteration. k_1 is given by this equation:

$$k_1 = \frac{-ln(final\_L/L_0)}{T} \tag{3}$$

Where final_L is the final learning rate (for the last iteration), which was set to 0.01, so k_1 comes out as 0.000461.

The neighbourhood radius is given by equation 4 below, from [8] :

$$r(t) = \begin{cases} (r_0 - 1) \cdot \left(1 - \frac{t}{t_1}\right) + 1 & \text{if } t \le t_1 \\ 1 & \text{if } t > t \end{cases} \quad (4)$$

where r is the radius, t is the number of iterations, r_0 is the initial radius, t_1 is a learning parameter valued at about a third of the number of iterations. This equation says that the neighbourhood radius will decrease linearly with iterations, until $t = t_1$, at which point the radius will be one node. The value of t_1 is set to $^1/_3$ of total iterations.

In Figure 2, we see the GUI (graphical user interface) for selecting the training parameters for HASSANN. The map length determines the size of the square map. A SOM with 40 as the map length would have 40 x 40 nodes.

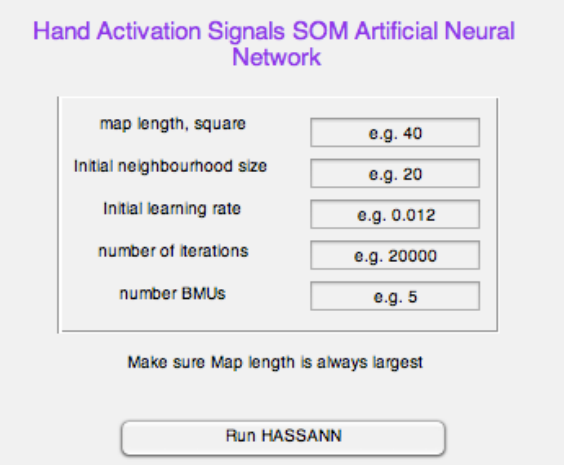


Figure 2: the GUI to select training parameters for HASSANN SOM.

Looking at Figure 2, we can see the initial neighbourhood size: an initial neighbourhood size of 20 would select a circle of nodes, with radius 20 nodes, around the BMU to have their MES spectrum made more similar to the training set spectrum selected at random from the set in a particular iteration.
The initial neighbourhood radius, is very important for clustering, as this defines which nodes will be trained to be more like the training set spectrum during each iteration of training. This radius decreases in size through the iterations to allow for increased fine-tuning of the clustering with later iterations.

The initial learning rate is called L_0 in equation (2), it is the first value by which the vectors in the nodes are updated in the first iteration if they are inside the radius mentioned above.

The number of iterations is linked to the initial learning rate: a higher L_0 means that the SOM requires a shorter training period, but the training period should be long enough that all nodes get updated/trained well.

Number of BMUs (Figure 1(c)) is how many MES spectra from the map nodes go into creation of the model spectrum, as can be seen in Figure 1(d). This gives the number of BMUs for the third stage of SOM: testing. The second stage of the SOM also has a BMU number for clustering the movement information; this is set in the code.

Results

To find parameters that would cause it to train well, HASSANN underwent multiple validation runs. Figure 1 shows the results from the 61st MES spectrum in the test run. Figure (a) is a surface plot of the MES spectra. It can be seen that the clustering looks reasonable from this figure of the 19th element in the feature vector. In Figure (b), we see the clustered movements. Figure (c) shows the BMU locations: the top-left corner is the location of the virtual MES spectrum that created the model plotted in Figure (d). Interestingly, it is far from any experimental, MES BMUs. The model plotted in (d) shows the experimental MES spectrum and the residual as well. The model has an NRMSD of 0.00939, which is extremely good: a value of 0.03 or lower is good.

The mean RMSD for 66 tests (6 repeats of 11 spectra) with the best parameters from validation was 0.0371 to 3 significant figures. These training parameters were as follows: the length of the square map was 40, the initial radius of the neighbourhood was 40, the initial learning rate was 0.1, the number of iterations used was 5,000, and the number of BMUs was 1. Additional parameters: the learning rate parameter k_1 was set to $4.61*10^{-4}$, and t_1 was set to a third of total iterations.

See Figure 2 for the HASSANN parameter selection, not all of the possible parameters are modifiable in this GUI; one would have to go into the code.

The confusion matrix for the test run with the above parameters is shown in Figure 3, below. The confusion matrix is a plot of actual classes on the vertical axis, versus classes perceived by HASSANN on the horizontal axis. The numbers range from 0.00 for 'in all tests this signal is not ever recognised as the movement on this square', to 1.00 for 'this signal is recognised as the movement on this square 100 % of the time'. So, ideally, there should be a diagonal from (1,1) to (11,11) that is entirely dark red (1.00), then everywhere else would be dark blue (0.00). This would show that the signal for *each* movement is recognised correctly in 100 % of tests.
This confusion matrix shows that there is mostly correct recognition, with a mean accuracy of 0.90 ± 0.08, with the standard deviation as the error.
Here the blue surrounding the red diagonal shows that neighbouring movements are not recognised as each other, as it should be. For example the point 7,6 shows that movement 6 is not recognised as movement 7. It is in fact recognised as movement 6 (correctly) 100 % of the time. This is why all other squares in that row are dark blue.

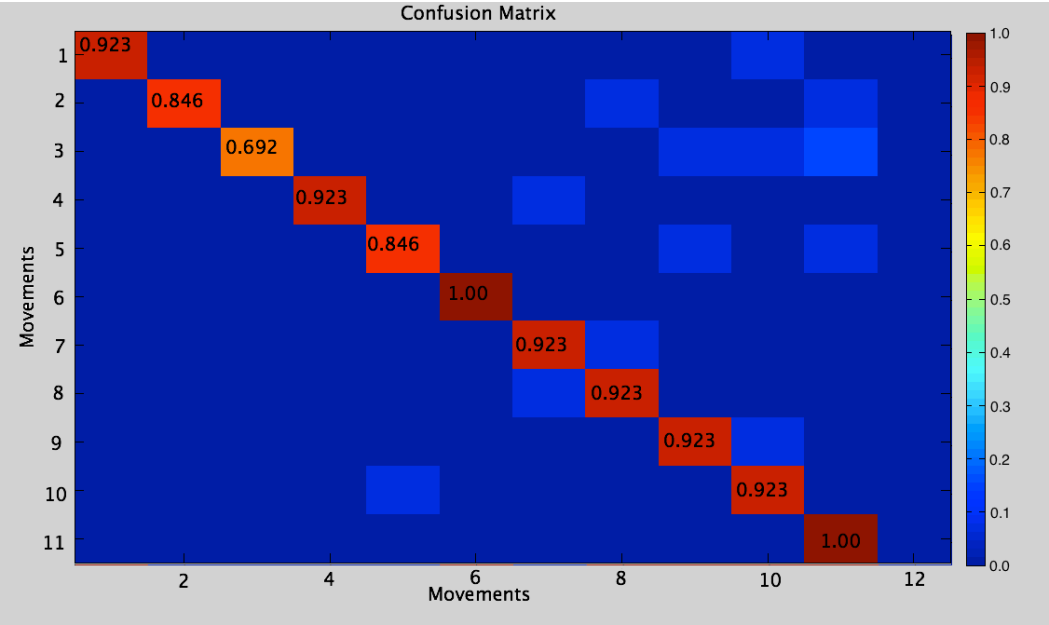


Figure 3: Confusion matrix from HASSANN; movements on vertical axis, predictions on horizontal axis. If the SOM were recognised the different movement classes perfectly, it would show a dark red line across the diagonal from top-left to bottom-right on the coordinates (1,1) through to (11,11). (The 12th column is just a plotting element.) This figure shows that each class is recognised correctly most of the time. The accuracies are shown in the diagonal squares. Mean accuracy for movement recognition is 0.90±0.08.

MES and corresponding movement classification

The overall mean of RMSE for the 66 test spectra is 0.037. The training set was 12x11, and the validation set was 6x11 in size.

Going back to Figure 1: this shows an example of a very good result. This is from the validation run 1, trial 61. The NRMSD of 0.00939 is very good, and we can see that the model (dashed blue) fits the experimental signal (black). The residual might be seen in faint, dotted red about the horizontal zero line. Figure (a) shows that the spectrum map has only 1 peak at this, the 19th element of the feature map. That shows there is only 1 feature that peaks at element 19, of 52 that make up the spectrum. This map looks well clustered. Figure (b) shows that the movements have all clustered in the corner set aside for movement 11, so this is successful. Figure (c) shows the BMUs of experimental MES and movements, these are mostly clustered at the bottom, far from the red U1 (top left) that represents the BMU of the model spectrum in Figure (d). Despite this, the model is very good.

A worse example of modelling MES spectra can be seen in Figure 4 below.

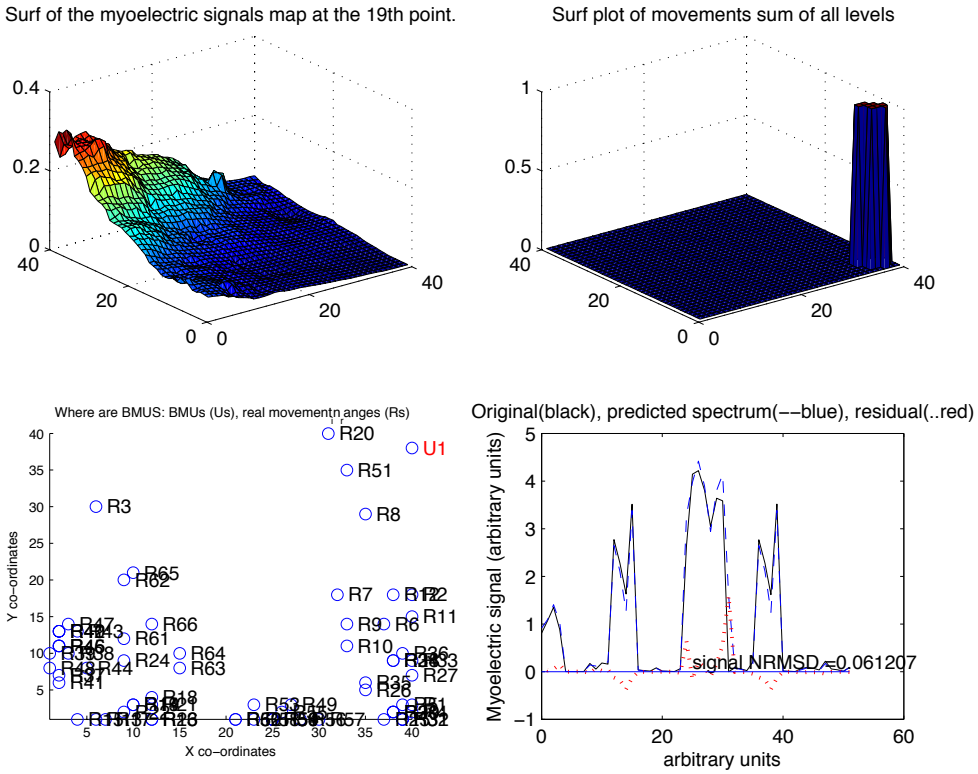


Figure 4: As Figure 1, except results from the 60th trial spectrum of the movement 11 (rest hand) validation run. Figure (d) shows that this model, with an NRMSD of 0.0612 is not as good that shown in Figure 1, but still reasonable.

Figure 4 shows worse results (than that of Figure 1), but the model (in Figure (d)) is still of reasonable quality. This is from validation run 1 as well, spectrum 60.
Here Figures (a) and (b) are the same as in Figure 1(a) and (b), as this is the same validation run. Figure (c) is very similar to that of Figure 1 as well, except the where the model (red ”U1”) is in a different position. This is one of the virtual spectra from the top right of the map, close to only a few of the experimental spectra. The RMSE of this validation run was 0.0371.

For different validation runs and tests, these maps are different each time, as the initial numbers populating them are pseudorandom. In Figure 5 in, the Discussion, it can be seen that the maps of Figures (a), (b) and (c) are different from those of Figures 1 and 4. That is because results shown in Figure 5 are from the test run, once the parameters were optimised.

Discussion

Next stage: Real-time testing

The next stage of testing would see the N subjects participate in the real-time evaluation for individual and simultaneous motions. Subjects would be both those who have

missing limbs, and those with fully, complete limbs. This testing would be done with 11 motions and 4 electrodes, this would be called HASSANN–RT, where RT stands for 'real time'. This has not been done, due to lack of time, with both authors writing up their PhD theses.
Thus far, HASSANN has only classified individual movements (one at a time) for BioPatRec. Another stage for HASSANN would be to recognise patterns in MES generated by simultaneous movements of the arm-wrist-hand prosthetic. These are when the subject moves their hand doing two or more of the movements at the same time, in the same manner that natural hands move: e.g. opening hand while moving to pronate (palm down).

Table 1: A summary of the MES data gathered from test subjects, DoF is degrees of freedom.

	Individual	Simultaneous	HASSANN-RT
Motions	11	27 (3 DoF)	11
Electrodes (bipolar)	4	8	4
Subjects	20	17	N

MES and corresponding movement classification

In Figure 1 (d), the NRMSD is extremely small, as the model MES spectrum is extremely good; this is the model for the movement 11, or "rest". Likely, the NRMSD is so good because the MES spectrum of the rest movement is quite different from other movements, and so classification is easier. This is probably also the reason the U1 BMU that the model comes from is so far from most experimental BMUs marked with R# on the BMU locations plot of Figure (c).

It is interesting that such good spectral NRMSDs can come from models made with just one node, or BMU. In ref [8] Hall and Rodger et *al*. found that protein's circular dichroism (CD) spectra were best when using 5 BMUs to make the model spectra. This is because the proteins all have different CD spectra. While the limb MES models are likely able to use just 1 BMU because the MES repeats from one type of movement should all be very similar if produced by the same muscles of the same person, using the same equipment.

For the sake of completeness, and to show how the maps are different each re-training, we have included results from the offline test run of HASSANN, see Figure 5. Figure 5(a) shows a very similar map to that of Figures 1 and 4 (a) (this is still feature map element 19 shown), but the coordinates are flipped around. The peak in Figure (b) is in the corner, as it should be, but the positions of this map do not correspond to those of Figures 1 and 4 (b).
Figure (c) map looks different as well, but the U1 BMU is still far from the R# BMUs. The BMUs have still clustered in a very similar way: there are a few large clusters, real spectra R7 – R11 are all close together, as in previous maps, R65, R51, R20 and R3 are all in parts of the map with sparse BMU coverage.

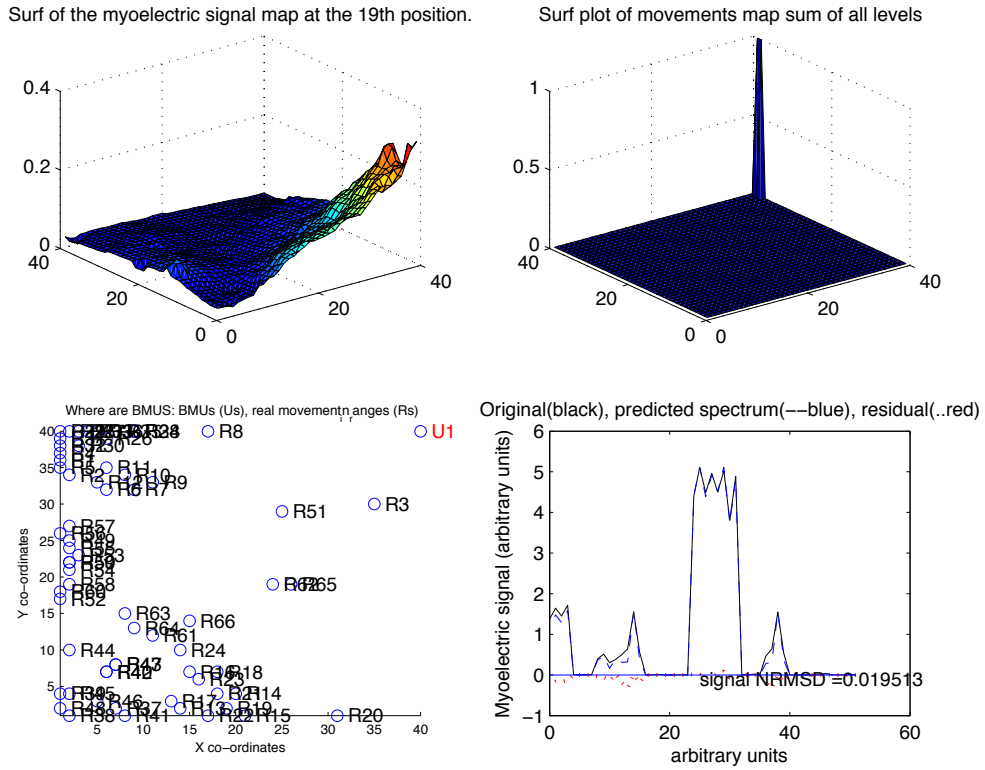


Figure 5: The results from the test run after the validation. Note how Figures (a), (b) and (c) are different from those above, as this is a separate map training.

This model MES spectrum looks good; it probably produced the correct movement prediction for the prosthetic limb. This model is of the rest movement; if we look at the confusion matrix, we see that the movement was predicted correctly 100 % of the time, so this model definitely led to a correct movement prediction.

Confusion Matrix

In Figure 3 in the Results, we see the confusion matrix from the offline test; here there is a good, red diagonal across all movements. Eight movements are recognised with 90 % or more accuracy, using this patient's data. This leaves only 3 movements that are not above this threshold.
The worst movement recognition is recognised correctly 69.2 % of the time, this is movement 3: flex hand. The second worse is shared between movements 2 and 5, or "close hand" and "pronation" both at 84.6 % accurate, the rest are very good, especially movements 6 and 11, supination (palm down) and rest.

It was noted above, that some movements produce MES that are very deterministic, while others are rather stochastic. Likely this is one reason some movements have bad accuracies, but certainly some of the reason is to do with training parameters that need to be optimised. Some of the reason is due to the pseudorandom numbers populating the initialised spectral map, and the pseudorandom selection of training spectra to compare with the map. These accuracies may also vary depending on which test subject the MES came from.

Perhaps the number of iterations, at 5000, was too small to allow all nodes to fully take on the forms of all of the MES. 5000 iterations and 40x40 = 1600 nodes.

Let us examine this, the experimental spectra are selected at random from the training set, and the number of times a node of the map is updated depends on the number of times it falls within the neighbourhood of the BMU.

We will use equation 5, below, which is the area of a circle rearranged, here for ¼ of the circle's area. This is because the smallest area is when the BMU is in the corner, so the area of the neighbourhood is ¼ of the area of a circle with radius r. Then rearranging the first part of equation 4 for t, to become equation 6, we can find when at least half the map should be selected, and at most all of the map's 1,600 nodes.

$$r = \sqrt{\frac{4a}{\pi}} \tag{5}$$

$$t = \left(1 - \left(\frac{r-1}{r0-1}\right)\right) t_1 \tag{6}$$

The area we want is at least 800 nodes (a = 800). If we enter these values, r comes out as 32 nodes. Using equation 6, with r_0 is 40 nodes, t_1 is 1,667 iterations (number of iterations divided by 3), we find that for t = 341 iterations (rounding down), at least half of the map is updated each iteration.

This is only an estimate, as, in these calculations, the circle of neighbourhood is defined as being a perfect circle, while in HASSANN's map it is a roughly circular shape made of squares. Nevertheless, we can get an idea of how many times the nodes are updated during training.
Considering that the learning rate is initially 0.1, and after 341 iterations has reduced slightly to 0.0854, see equations 2 and 3, with a k_1 value of 0.0002. This means each node that *always* falls in the neighbourhood region is made similar to real MES spectra 341*0.0854 = at least 29 times (rounding down). (A learning rate of 1.00 would immediately make a node exactly the same as the training spectrum, but this would be no good for clustering.)
So, we can be reasonably sure that the nodes are updated sufficiently, even noting that few would fall within the neighbourhood *every* iteration.

The corner and edge nodes would be the locations of the most extreme spectra, and in the BMU locations maps of Figures (c) in 1, 4, and 5, at least seven clusters can be made out. The SOM clearly knows where the effective borders of the full set of 11 clusters are, this can be seen from the accuracy rates.

For the SOM to be acceptable for real-time limb control the diagonal from top-left to bottom-right would have to be complete and very clear, as the movements need to be recognised as what they are intended to be with a very high accuracy.

Work by Ortiz-Catalan et *al.* reported on in reference [12] shows that the accuracies of pattern recognition algorithms in offline tests are usually significantly higher that the real-time accuracies for the same systems. In the real-time tests reported on in that paper, people used the pattern recognition algorithm in question, as part of BioPatRec, to control actual robotic arms. Ortiz-Catalan et *al.* reported the following for LDA: offline mean accuracy 92.1 ± 4 %, real-time mean accuracy 67.1 ± 10 %. There were similar values for MLP and RFN, so there is a significant step down in accuracy over all of those algorithms.

Conclusion

HASSANN, the self–organising map derived from SSNN, the SOM developed for finding protein secondary structures from circular dichroism, has been adapted for the pattern recognition stage of real-time robotic prosthetic, limb control software benchmarking platform, BioPatRec.
The HASSANN SOM methodology has attained the accuracy value of 0.90 ± 0.08 for offline tests with the 11 hand and wrist movements (10 movements plus rest). The next stage is for HASSANN to perform the real-time test with full-limbed, and disabled people with robotic, prosthetic arms. After that, HASSANN will be trained with simultaneous prosthetic hand movements for more natural control, and later tested in real-time.

Acknowledgements

We thank the Engineering and Physical Sciences Research Council for the funding for Vincent Hall through the MOAC Doctoral Training Centre (Grant number EP/F500378/1).

References

1. Brenner CD, Brenner, J. K. The Use of Preparatory/Evaluation/Training Prostheses in Developing Evidenced-Based Practice in Upper Limb Prosthetics. Journal of Prosthetics & Orthotics 2008;20(3):70-82.
2. Ortiz-Catalan M, Branemark, R., Hakansson, B. Evaluation of Classifier Topologies for the Real-time Classification of Simultaneous Limb Motions. 2013; Osaka, Japan. IEEE.
3. Dictionary TF. http://www.thefreedictionary.com.
4. NHS Choices your health yc. Amputation. Volume 2014; 2012.
5. Le Blanc M. "Give Hope – Give a Hand" – The LN-4 Prosthetic Hand. Volume 2014; 2008.
6. Farry KA, Walker, I. D., Baraniuk, R. G. Myoelectric Teleoperation of a Complex Robotic Hand. IEEE Transactions on Robotics and Automation 1996;12(5):775–788.
7. Carrozza MC, Cappiello, G., Micera, S., Edin, B. B., Beccai, L., Cipriani, C. Design of a cybernetic hand for perception and action. Biological Cybernetics 2006;95:629–644.
8. Hall V, Nash, A., Hines, E., Rodger, A. Elucidating protein secondary structure with circular dichroism and a neural network. Journal of Computational Chemistry 2013;34(32):2774-2786.
9. Hall V, Nash, A., Rodger, A. SSNN, a method for neural network protein secondary structure fitting using circular dichroism data. Submitted to Analytical Methods; 2014.
10. Hall V, Sklepari, M., Rodger, A. Protein Secondary Structure Prediction from Circular Dichroism Spectra Using a Self–organising Map with Concentration Correction. Chirality 2014.

11. Ortiz-Catalan M. https://code.google.com/p/biopatrec/downloads/list. 2013. p BioPatRec Downloads.
12. Ortiz-Catalan M. BioPatRec: A modular research platform for the control of artificial limbs based on pattern recognition algorithms. Source Code for Biology and Medicine 2013;8(11).
13. Huang Y, Engelhart, K. B., Hudgins, B., Chan, A. D. C. A Gaussian Mixture Model Based Classification Scheme for Myoelectric Control of Powered Upper Limb Prostheses. IEEE Transactions on Biomedical Engineering 2005;52(11):1801-1811.
14. Kyberd PJ, Holland, O. E., Chappell, P. H., Smith, S., Tregidgo, R., Bagwell, P. J., Snaith, M. MARCUS: A two degree of free- dom hand prosthesis with hierarchical grip control. IEEE Transactions on Rehabilitation Engineering 1995;3(1):70–76.
15. Ortiz-Catalan M. http://www.youtube.com/user/NCALOI. Volume 2014. YouTube; 2012.
16. Hudgins B, Parker, P., Scott, R. N. A New Strategy for Multifunction Myoelectric Control. IEEE Transactions on Biomedical Engineering 1993;40(1):82–94.
17. Chu J-U. M, I., Mun, M-S. A Real-Time EMG Pattern Recognition System Based on Linear-Nonlinear Feature Projection for a Multifunction Myoelectric Hand. IEEE Transactions on Biomedical Engineering 2006;53(11):2232-2239.
18. Schmidhuber J. Learning complex, extended sequences using the principle of history compression. Neural Computation 1992;4(2):234–242.
19. Hopkins PM. Skeletal muscle physiology. Contin. Educ. Anaesth. Crit. Care Pain 2006;6(1):1-6.
20. Parker PA, Stuller, J. A., Scott, R. N. Signal Processing for the Multistate Myoelectric Channel Proceedings of the IEEE 1977:662–674.
21. Ortiz-Catalan M. https://code.google.com/p/biopatrec/wiki/BioPatRec. Volume 2014; 2014.
22. Li G, Schultz, A. E., Kuiken, T. Quantifying pattern recognition-based myoelectric control of multifunctional transradial prostheses. IEEE Trans Neural Syst Rehabil Eng 2010;18(2):185–192.
23. Ortiz-Catalan M, Branemark, R., Hakansson, B. Biologically inspired algorithms applied to prosthetic control. 2012 15-17 Feb.; Innsbruck. p 7–15.
24. Hall V, Sklepari, M., Rodger, A. Protein Secondary Structure Prediction from Circular Dichroism Spectra Using a Self-organising Map with Concentration Correction. Chirality 2014.
25. Kohonen T. Self-Organised Formation of Topologically Correct Feature Maps. Biological Cybernetics 1982;43:59–69.